

**J. Ángel Velázquez Iturbide**

# **Un Estudio de la Creatividad Algorítmica de los Alumnos**

**Número 2020-02**

**Serie de Informes Técnicos DLSI1-URJC**

**ISSN 1988-8074**

**Departamento de Lenguajes y Sistemas Informáticos I**

**Universidad Rey Juan Carlos**



## Índice

1	Introducción .....	1
2	Metodología .....	2
2.1	Contexto y Protocolo.....	2
2.1	Análisis.....	3
3	Resultados de la Práctica 2 – Algoritmos Heurísticos.....	4
3.1	Entregas de los Alumnos.....	4
3.2	Resultados de Fluidez.....	10
3.3	Categorías de Algoritmos.....	10
3.3.1	Algoritmos Basados en el Enunciado.....	11
3.3.2	Algoritmos con Recorrido Lineal.....	15
3.2.4	Algoritmos con una Ordenación Inicial .....	16
3.3	Resultados de Flexibilidad .....	17
3.4	Resultados de Originalidad .....	18
3.5	Resultados Globales de Creatividad.....	18
3.6	Debate .....	19
4	Resultados de la Práctica 3b – Ramificación y Poda.....	20
4.1	Entregas de los Alumnos.....	20
4.2	Resultados de Fluidez.....	23
4.3	Categorías de Algoritmos.....	23
4.4	Resultados de Flexibilidad .....	24
4.5	Resultados de Originalidad y Elaboración .....	24
4.6	Resultados Globales de Creatividad.....	26
4.7	Debate .....	27
5	Resultados de la Práctica 6 – Algoritmos Probabilistas .....	28
5.1	Entregas de los Alumnos.....	28
5.2	Resultados de Fluidez.....	30
5.3	Categorías de Algoritmos.....	30
5.4	Resultados de Flexibilidad .....	31
5.5	Resultados de Originalidad y Elaboración .....	31
5.6	Resultados Globales de Creatividad.....	33
5.7	Debate .....	34
6	Debate .....	35
5	Conclusiones .....	36
	Agradecimientos. ....	36
	Referencias.....	36
	Apéndice A: Enunciado de las Prácticas.....	38



# Un Estudio de la Creatividad Algorítmica de los Alumnos

J. Ángel Velázquez Iturbide

Departamento de Informática y Estadística, Universidad Rey Juan Carlos,  
C/ Tulipán s/n, 28933, Móstoles, Madrid  
angel.velazquez@urjc.es

**Resumen.** Con frecuencia, se asocia el pensamiento algorítmico con la creatividad. Hemos evaluado la creatividad de los alumnos en el diseño de técnicas algorítmicas: algoritmos heurísticos para un problema de optimización, función de cota para un algoritmo de ramificación y cota, y algoritmos probabilistas para un problema de optimización. En los dos primeros casos se ha usado el mismo problema de optimización. La creatividad se ha medido mediante la fluidez, flexibilidad, originalidad y elaboración mostrada por los alumnos. El informe presenta los resultados obtenidos en cada caso, así como reflexiones metodológicas sobre la medida de la creatividad y la relevancia de los resultados.

**Palabras clave:** Algoritmos, creatividad, análisis de contenidos.

## 1 Introducción

Desde hace años, venimos diseñando y evaluando herramientas software y métodos didácticos sobre la docencia de algoritmos. Se han evaluado distintos aspectos, como el rendimiento de los alumnos (mediante la experimentación interactiva con algoritmos voraces [1][2], el uso de la técnica de “múltiples visualizaciones” para la comprensión de los algoritmos de programación dinámica [3], o el uso de tests y “ejemplos trabajados” para toda la asignatura [4]), malasconcepciones de los alumnos (sobre algoritmos voraces [5][6] o en general sobre algoritmos de optimización [7][8]) y motivación de los alumnos (por el uso del sistema de visualización SRec para la eliminación de la recursividad redundante [9]).

Con frecuencia, la resolución algorítmica de problemas se asocia con la creatividad. En este informe se presenta una evaluación de la creatividad de los alumnos en una asignatura de algoritmos. La estructura del informe es la siguiente. El apartado 2 describe el contexto educativo y los instrumentos de análisis utilizados en la evaluación. Los apartados 3 al 5 presentan los resultados obtenidos en cada técnica de diseño de algoritmos. En el apartado 6 se debaten los resultados y los métodos de análisis. Finalmente, se presentan las conclusiones del estudio y un apéndice con los enunciados de las prácticas analizadas.

## 2 Metodología

### 2.1 Contexto y Protocolo

Este estudio se realiza dentro de la asignatura optativa “Algoritmos Avanzados”, de cuarto curso del Grado en Ingeniería Informática (GII). La asignatura también se ofrece a los alumnos del Grado en Ingeniería de Computadores (GIC) y a los alumnos del grado doble GII-GIC. Todos estos alumnos asisten al mismo grupo presencial, por lo que no los desglosamos en el análisis presentado en este informe. Por último, la asignatura se ofrece a un segundo grupo, formado por alumnos de GII online, en extinción. La evaluación se ha realizado durante el periodo septiembre-diciembre del curso académico 2019-20.

El temario de la asignatura incluye varias técnicas algorítmicas para la resolución de problemas de optimización (técnica voraz, algoritmos heurísticos y aproximados, vuelta atrás, ramificación y poda, y programación dinámica), así como algoritmos probabilistas (con un ámbito distinto).

Se propuso la realización de 6 prácticas al grupo presencial y sólo las 5 primeras al grupo online:

1. Técnica voraz. Dado un problema de planificación de tareas [10, págs. 392-393] y una descripción de un criterio voraz, se pide desarrollar el algoritmo sin presuponer que las actividades ya vienen ordenadas.
2. Algoritmos heurísticos. El enunciado plantea un problema de planificación de trabajos con beneficio máximo [11, pág. 313] y esboza un algoritmo heurístico. Se pide especificar el problema, programar dos algoritmos heurísticos al menos (el esbozado y otro más como mínimo), y comprobar su optimalidad con OptimEx.
3. Técnicas de búsqueda. La práctica se realizó con dos entregas:
  - a. Dado el mismo problema de planificación de trabajos, se pide desarrollar un algoritmo de vuelta atrás y comparar la optimalidad de este algoritmo y de los algoritmos heurísticos desarrollados en la práctica 2.
  - b. Dado el mismo problema de planificación de trabajos, se pide desarrollar un algoritmo de ramificación y poda, y comparar la optimalidad y los tiempos de ejecución de este algoritmo y de los anteriormente desarrollados.
4. Eliminación de recursividad múltiple redundante. Se da una función recursiva  $f$  y se pide analizar su redundancia, eliminarla mediante memorización y tabulación, y analizar la complejidad en tiempo y memoria de ambos algoritmos.
5. Técnica de programación dinámica. Dado el mismo problema de planificación de trabajos de las prácticas 2 y 3, se pide desarrollar un algoritmo recursivo, convertirlo en un algoritmo de programación dinámica (tabulador), analizar su complejidad, y comparar la optimalidad y los tiempos de ejecución de ambos algoritmos y de los anteriormente desarrollados.

6. Algoritmos probabilistas. Dada la versión de minimización del problema de llenado de cajas [12, págs. 534-535], se esboza un algoritmo probabilista y se pide comparar su optimalidad con la de otros algoritmos para el mismo problema disponibles en el Aula Virtual.

Las prácticas podían realizarse individualmente o en pareja. Cada grupo de prácticas debía entregar un informe, siguiendo un índice especificado en el enunciado de cada práctica.

Una vez que una práctica estaba corregida, en el plazo de varios días se publicaba su calificación junto con comentarios sobre la misma. Los alumnos podían aprovechar estos comentarios para volver a entregarla en un breve plazo, aunque con una nueva nota máxima menor de 10. Por esta razón, algunos grupos han realizado dos entregas de algunas prácticas.

Las prácticas 1 y 4 tienen como objetivo ejercitarse en el uso correcto de alguna técnica. En ellas no se debe crear un algoritmo original sino transformar un algoritmo en otro relacionado. Por tanto, en estas prácticas se valora más el rigor que la creatividad.

Las restantes prácticas se prestan más a la originalidad ya que tienen como objetivo aplicar alguna técnica de diseño de algoritmos, desarrollando un algoritmo. No obstante, tampoco analizamos la creatividad en las prácticas 3a y 5: en la primera no se solicitó proponer varios árboles de búsqueda y en la segunda, el margen de variación se limita principal a una formulación recursiva hacia adelante o hacia atrás.

Por último, las prácticas 2, 3b y 6 se prestan a una creatividad limitada, normalmente en algún elemento de diseño. Analizamos su creatividad en los tres apartados siguientes. Incluimos los enunciados de las prácticas 2, 3a, 3b y 6 en el Apéndice.

## 2.1 Análisis

El análisis de la creatividad de los alumnos se ha realizado en el marco de la teoría de la creatividad de Torrance [14]. Pueden distinguirse cuatro componentes de la creatividad:

- **Fluidez.** Se asocia con el número de respuestas o soluciones dadas ante una situación.
- **Flexibilidad.** Se asocia con cambios de orientación en las soluciones generadas.
- **Originalidad.** Se asocia con la innovación y carácter único de una o varias soluciones.
- **Elaboración.** Se asocia con el grado de desarrollo de cada solución.

Cada componente puede medirse de forma distinta:

- **Fluidez.** Es igual al número de algoritmos presentados.
- **Flexibilidad.** Depende de cada caso. En líneas generales, los algoritmos se analizan de forma cualitativo y se clasifican en categorías adecuadas para su técnica de diseño asociada. La flexibilidad de un grupo de prácticas corresponderá al número de categorías distintas de sus algoritmos.

- Originalidad. Si el profesor da un algoritmo como orientación, la originalidad de un algoritmo sería proporcional a lo distinto que sea con respecto a la orientación del profesor. En caso de no haber ninguna orientación, la originalidad de un algoritmo es inversamente proporcional a la frecuencia de aparición del mismo algoritmo, o similares.
- Elaboración. Consideramos que un algoritmo está elaborado completamente si se ha programado, siendo menor pero discutible su grado de elaboración si solamente se ha “ideado”.

También se mide la creatividad total mostrada por cada grupo. Aunque varía en cada caso, es proporcional a la suma de la creatividad asociada a cada algoritmo, ponderada según la flexibilidad mostrada en los algoritmos propuestos. La creatividad de cada algoritmo tiene en cuenta su originalidad y elaboración.

En el análisis de cada práctica, se detalla cómo se mide cada componente de la creatividad y la creatividad total. También se presenta el análisis cualitativo de las aportaciones, que sirve para determinar cómo medir la flexibilidad de cada una originalidad. En el apartado 6 se debaten las mediciones realizadas.

En los próximos apartados analizamos sucesivamente el análisis de creatividad de las prácticas 2, 3a, 3b, 5 y 6.

### 3 Resultados de la Práctica 2 – Algoritmos Heurísticos

El enunciado de la práctica 2 esbozaba un algoritmo heurístico H1, según el cual se eligen trabajos mediante la comparación sucesiva, desde  $i=0$  y cada dos semanas  $i$  e  $i+1$ , entre los ingresos  $b_i+b_{i+1}$  y  $a_{i+1}$ .

A continuación, se presentan sucesivamente las entregas de los alumnos, resultados de fluidez, análisis cualitativo de los algoritmos heurísticos, resultados de flexibilidad, resultados de originalidad y creatividad global.

#### 3.1 Entregas de los Alumnos

Se recogieron 34 informes de prácticas, 2 del grupo online y 32 del grupo presencial. De ellas, 25 correspondían a parejas de alumnos (1 del grupo online y 24 del grupo presencial) y 9 a alumnos individuales (1 del grupo online y 8 del grupo presencial).

En la Tabla 1 incluimos las entregas recibidas; sólo se analiza la primera entrega ya que la segunda entrega estaba condicionada por los comentarios del profesor. Se presenta el número de cada grupo de prácticas, el grupo de clase al que pertenecen, si era una pareja o un alumno solo, una breve descripción de las características de cada algoritmo heurístico desarrollado, la categoría en que clasificamos cada algoritmo según su similitud al algoritmo esbozado (véase Apartado 3.3), el total de originalidad del grupo, y el porcentaje de casos de prueba para el que cada algoritmo ha calculado un resultado óptimo. Este último dato depende de los materiales (algoritmos) y las condiciones del experimento (número de casos de prueba, longitud y contenido de los casos de prueba); en la columna incluimos el número de casos de prueba y su longitud.



Tabla 1. Informes de prácticas presentados para la práctica 2

Nº grupo	Grupo de clase	Indiv./ pareja	Algoritmos	Categorías	Originalidad	Resultados experimentales
1.	OL	I	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Alterna trabajos de baja tensión, descanso y alta tensión desde la primera semana con alta tensión. Incrementos de 3 3. Sólo toma trabajos de alta tensión en semanas pares (desde la primera), salvo quizá en la última semana impar. Incrementos de 2 4. Elige entre todos los trabajos pares de baja o de alta tensión, salvo quizá en la última semana impar 5. (H1) Contempla la primera semana. Incrementos de 2	1. Enunciado 2. Lineal 3. Lineal 4. Lineal 5. Enunciado	7	(N=1.000, S=12) 1: 24'6% 2: 0'01% <b>3: 0%</b> 4: 5'9% 5: 90'9%
2.	OL	P	1. Similar a H1. No contempla la primera semana. Incrementos de 1. Incumple restricciones: puede sumarse $b[i]$ 2 veces o $b[i]+a[i+1]$ 2. Similar a H1. No contempla la primera semana. Incrementos de 1. Incumple restricciones: puede sumarse $a[i+1]$ sucesivas veces	1. Enunciado 2. Enunciado	1	(N=1.000, S=5) 1: 60'5% <b>2: 100%</b>
3.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Elige entre todos los trabajos de baja o de alta tensión. Contempla elegir un trabajo de alta tensión en la primera semana pero no coger ninguna en la última semana si hay un número impar	1. Enunciado 2. Lineal	2	(N=100, S=12) <b>1: 100%</b> 2: 28%
4.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Similar a H1. Contempla la primera semana. Incrementos de 3 3. Elige entre todos los trabajos de baja tensión, pares de alta tensión o impares de alta tensión 4. (H1) Contempla la primera semana. Incrementos de 2	1. Enunciado 2. Enunciado 3. Lineal 4. Enunciado	4	(N=100, S=10) 1: 13% 2: 16% 3: 24% 4: 72%
5.	Pr	P	1. Similar a H1. Contempla la primera semana. Incrementos de 1. Incumple restricciones: puede sumarse $a[i+1]$ anticipadamente 2. Sólo toma trabajos de baja tensión 3. Elige entre los trabajos pares de alta tensión o impares de alta tensión	1. Enunciado 2. Lineal 3. Lineal	4	(N=100, S=5) 1: 96% 2: 47% 3: 4%
6.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 1 2. Alterna trabajos de baja tensión, descanso y alta tensión desde la primera semana con alta tensión. Incrementos de 1, con un contador auxiliar 3. (H1) Contempla la primera semana. Incrementos de 2	1. Enunciado 2. Lineal 3. Enunciado	3	(N=100, S=15) 1: 30% <b>2: 0%</b> 3: 75%

7.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Similar a H1. Contempla la primera semana. Incrementos de 1, con una variable auxiliar	1. Enunciado 2. Enunciado	1	(N=100, S=6) 1: 46% 2: 95%
8.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2: Similar a H1. No contempla la primera semana. Incrementos de 1. Permite varios descansos sucesivos para ganar un beneficio alto	1. Enunciado 2. Enunciado	1	(N=100, S=5) 1: 73 2: 99
9.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. (H1) Contempla la primera semana. Incrementos de 2 3. Elige entre trabajar sólo días pares o sólo días impares, eligiendo siempre el trabajo de mayor beneficio en la semana	1. Enunciado 2. Enunciado 3. Lineal	3	(N=200, S=13) 1: 26% 2: 77% 3: 0'5%
10.	Pr	I	1. (H1) Contempla la primera semana. Incrementos de 2 2. Elige entre todos los trabajos de baja tensión, pares de alta tensión o impares de alta tensión	1. Enunciado 2. Lineal	2	(N=100, S=5) 1: 53% 2: 76% (N=100, S=20) <b>1: 100%</b> 2: 32% (N=100, S=20) 1: 66% 2: 53%
11.	Pr	I	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. (H1) Contempla la primera semana. Compara sucesivamente un trabajo de baja tensión y uno de alta, descanso y uno de alta, y dos trabajos de baja tensión. Incrementos de 1	1. Enunciado 2. Enunciado	1	(N=100, S=20) 1: 20% 2: 97%
12.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos variables de 1 (trabajos de baja tensión) ó 2 (de alta) 2. Se ordenan los beneficios altos y se van eligiendo todos los posibles, manteniendo las restricciones del problema	1. Enunciado 2. Ordenación	3	(N=100, S=7) 1: 77% 2: 25%
13.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 1+1 2. Elige entre todos los trabajos de baja o de alta tensión, contando sólo los trabajos de alta tensión mayores que los respectivos de baja tensión	1. Enunciado 2. Lineal	2	(N=1.000, S=10) <b>1: 100%</b> 2: 90'1%
14.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos variables de 1 (trabajos de baja tensión) ó 2 (de alta) 2. Variante de la anterior, donde sólo se comparan $a[i+1]$ y $b[i+1]$	1. Enunciado 2. Enunciado	1	(N=200, S=11,12) 1: 92% 2: 9'5%

15.	Pr	P	1: Similar a H1. No contempla la primera semana. Incrementos de 2 2: Similar a H1. Contempla la primera semana (de forma algo diferente). Incrementos variables de 1 (trabajos de baja tensión) ó 2 (de alta) 3: Variante de la anterior, donde también se comparan trabajos de tres semanas consecutivas. Incrementos variables de 1 (trabajos de baja tensión), 2 (de alta) ó 3 (trabajo de baja tensión en semana 1 y alta tensión en semana 3)	1. Enunciado 2. Enunciado 3. Enunciado	1	(N=1.000, S=12) 1: 43'2% 2: 72'2% 3: 16'8%
16.	Pr	I	1: Similar a H1. No contempla la primera semana. Incrementos de 2 2: Elige entre todos los trabajos de baja tensión, pares de alta tensión o impares de alta tensión	1. Enunciado 2. Lineal	2	(N=100, S=5) 1: 88% 2: 73%
17.	Pr	P	1: Similar a H1. No contempla la primera semana. Incrementos de 2 2: (H1) Contempla la primera semana. Incrementos de 2 3: Tomar los trabajos de alta tensión de las semanas pares	1. Enunciado 2. Enunciado 3. Lineal	3	(N=10, S=10) 1: 40% <b>2: 100%</b> <b>3: 0%</b> (N=10, S=9) 1: 40% <b>2: 100%</b> 3: 10%
18.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Sólo toma trabajos de baja tensión, salvo quizá en la primera semana 3. (H1) Contempla la primera semana. Incrementos de 2 4. Sólo toma trabajos de alta tensión en semanas pares, salvo quizá en la última semana impar	1. Enunciado 2. Lineal 3. Enunciado 4. Lineal	5	(N=100, S=8) 1: 28% 2: 24% 3: 78% 4: 3%
19.	Pr	I	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. (H1) Contempla la primera semana. Incrementos de 2 3. Recorrido en orden inverso comparando a[i] y b[i]. Decrementos variables de 1 (trabajos de baja tensión) ó 2 (de alta) 4. Similar a H1 y al anterior. Recorrido en orden inverso. Decrementos variables de 1 (trabajos de baja tensión) ó 2 (de alta)	1. Enunciado 2. Enunciado 3. Enunciado 4. Enunciado	3	(N=2.000, S=55) 1: 6'35% 2: 11'3% <b>3: 0%</b> 4: 84'3%
20.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Similar a H1. Contempla la primera semana. Incrementos de 1. Incumple restricciones: puede sumarse a[i+1] sucesivas veces 3. Sólo toma trabajos de baja tensión, salvo quizá en la última semana	1. Enunciado 2. Enunciado 3. Lineal	3	(N=1.000, S=10) 1: 65'6% 2: 94% 3: 0'1%

21.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. (H1) Contempla la primera semana. Incrementos de 2	1. Enunciado 2. Enunciado	1	(N=1.000, S=10) 1: 54.5% 2: 100%
22.	Pr	I	1. Similar a H1. No contempla la primera semana. Incrementos variables de 1 (trabajos de baja tensión) ó 2 (de alta) 2. Similar a H1. Contempla la primera semana. Incrementos variables de 1 (trabajos de baja tensión) ó 2 (de alta) 3. Parece igual al anterior	1. Enunciado 2. Enunciado 3. Enunciado	2	–
23.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Variante de la anterior donde se comparan $a[i]+b[i+1]$ , $a[i+1]$ y $b[i]+b[i+1]$ , teniendo en cuenta los descansos. Incrementos de 1	1. Enunciado 2. Enunciado	1	(N=100, S=12) 1: 30% 2: 96%
24.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Variante de la anterior donde se comparan $a[i]$ , $a[i+1]$ y $b[i]+b[i+1]$ , teniendo en cuenta los descansos. Incrementos de 2 3. Variante de la anterior donde se comparan $a[i]+b[i+1]$ , $a[i]+a[i+2]$ , $b[i]+a[i+2]$ y $b[i]+b[i+1]+b[i+2]$ , teniendo en cuenta los descansos. Incrementos de 3	1. Enunciado 2. Enunciado 3. Enunciado	2	(N=100, S=12) 1: 49% 2: 54% 3: 75%
25.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Variante de la anterior, implementado de forma complicada (se comprueba si merece la pena elegir un trabajo de alta tensión aunque suponga deshacer la elección del trabajo anterior). Contempla la primera semana. Incrementos de 1	1. Enunciado 2. Enunciado	1	(N=1.000, S=5) 1: 47.9% 2: 99.1%
26.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Variación del anterior. Compara sucesivamente $b[i]+b[i+1]$ , $a[i]+b[i+1]$ y $a[i+1]$ . Incrementos de 2 3. (H1) Contempla la primera semana. Incrementos de 2	1. Enunciado 2. Enunciado 3. Enunciado	2	(N=100, S=12) 1: 27% 2: 53% 3: 64%
27.	Pr	I	1. (H1) Contempla la primera semana. Incrementos de 2 2. Variación del anterior. Compara también $a[i+1]$ con $a[i+2]$ pero, quizá por error, sólo suma beneficios en el último mes. Incrementos de 1 3. Similar a H1. Hace comparaciones de $b[i]+b[i+1]$ con $a[i+1]$ pero, quizá por error, sólo suma beneficios en el último mes. Incrementos de 1	1. Enunciado 2. Enunciado 3. Enunciado	2	(N=100, S=12) 1: 100% 2: 0% 3: 0%
28.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Similar a H1. No contempla la primera semana. Incrementos de 1	1. Enunciado 2. Enunciado	1	(N=100, S=6) 1: 36% 2: 96%
29.	Pr	P	1. (H1) Contempla la primera semana, pero teniendo en cuenta también la segunda. Incrementos de 1, con variables auxiliares 2. (H1) Contempla la primera semana. Incrementos de 1, con variables auxiliares	1. Enunciado 2. Enunciado	1	(N=100, S=6) 1: 98% 2: 96%

30.	Pr	P	1. Similar a H1. Contempla la primera semana, al comparar $a[i]$ , $a[i+1]$ y $b[i]+b[i+1]$ . Incrementos de 1+1 2. Similar a H1. Contempla la primera semana. A veces, quizá por error, sólo compara $a[i+1]$ y $b[i+1]$ . Incrementos de 1 y variables auxiliares	1. Enunciado 2. Enunciado	1	(N=5, S=12) 1: 80% 2: 20%
31.	Pr	I	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Elige entre todos los pares o impares de alta tensión 3. (H1) Contempla la primera semana. Incrementos de 1	1. Enunciado 2. Lineal 3. Enunciado	3	(N=100, S=7) 1: 12% 2: 63% 3: 80%
32.	Pr	I	1. (H1) Contempla la primera semana, manipulando el índice $i$ . Incrementos de 2 2. Ordenación de los distintos trabajos y sucesivos recorridos de los vectores	1. Enunciado 2. Ordenación	3	(N=100, S=12) 1: 83% 2: 56%
33.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 1+1 2. Elige entre todos los trabajos de baja tensión, pares de alta tensión o impares de alta tensión	1. Enunciado 2. Lineal	2	(N=100, S=10) <b>1: 100%</b> 2: 34% (N=500, S=10) <b>1: 100%</b> 2: 44'4%
34.	Pr	P	1. Similar a H1. No contempla la primera semana. Incrementos de 2 2. Similar a H1. Contempla la primera semana mediante un recorrido inverso. Decrementos de 2	1. Enunciado 2. Enunciado	1	(N=1.000, S=15) 1: 21'9% 2: 85'1%

### 3.2 Resultados de Fluidez

La Tabla 2 muestra los resultados de fluidez, mostrando el número y porcentaje de grupos que han presentado distinto número de algoritmos. Recuérdese que el número mínimo de algoritmos a presentar era 2.

**Tabla 2.** Resultados de fluidez en la práctica 2 (N=34)

Número de algoritmos	# grupos	% grupos
2	19	56%
3	11	32%
4	3	9%
5	1	3%

Puede observarse que solamente un 44% de los grupos propusieron algún algoritmo más de los obligatorios, es decir, 3 o más. Incluso en estos casos, normalmente propusieron sólo un algoritmo adicional (32% frente a 12% de dos o tres algoritmos adicionales).

### 3.3 Categorías de Algoritmos

Los algoritmos desarrollados por los alumnos son de diversas clases. Muchas veces un grupo presentaba una interpretación del algoritmo esbozado y una variante parecida, con solamente diferencias menores. Sin embargo, en otros casos, el algoritmo tenía un enfoque distinto.

Para determinar de forma más precisa sus características, se realizó un análisis cualitativo bastante detallado. El proceso seguido fue el siguiente:

- Corrección de las prácticas. Se tuvo la intuición de que había tres clases de algoritmos: variantes del algoritmo esbozado en el enunciado, algoritmos “prometedores” y algoritmos “no prometedores”.
- Relleno de la Tabla 1. Al examinar cada práctica para resumir cada algoritmo y escribirlo en la tabla (columna cuarta), se llegó a la percepción de que había variantes del algoritmo esbozado en el enunciado, algoritmos basados en el cómputo exclusivo de trabajos de baja o alta tensión, y otros algoritmos.
- Iteración de etiquetado. Al rellenar la columna quinta (antepenúltima) de la tabla, se vio la necesidad de distinguir categorías más precisas. Aunque durante el proceso, se barajaron varias formas de concretar las categorías, finalmente quedaron las siguientes: algoritmos basados en el enunciado, algoritmos con un recorrido lineal y algoritmos basados en ordenación.

A continuación, se presentan las categorías y sus variantes resultantes del análisis cualitativo.

### 3.3.1 Algoritmos Basados en el Enunciado

El enunciado de la práctica (véase Apéndice) explica las restricciones que deben cumplirse para realizar trabajos de alta o de baja tensión. También esboza los principios para un algoritmo heurístico. Una forma de concretar dicho algoritmo, teniendo en cuenta las restricciones del problema, es:

```
public static int trabajosH2 (int[] as, int[] bs) {
    int benef = Math.max(as[0],bs[0]);
    int i;
    for (i=1; i<bs.length-1; i+=2) {
        if (bs[i]+bs[i+1] > as[i+1])
            benef += bs[i]+bs[i+1];
        else
            benef += as[i+1];
    }
    if (i==bs.length-1)
        benef += bs[i];
    return benef;
}
```

Obsérvese que el algoritmo consta de tres partes: primer mes, meses intermedios y último mes. La parte primera puede tenerse en cuenta a partir del enunciado del problema. Sin embargo, la parte segunda implementa el esbozo de algoritmo heurístico y la parte tercera es necesaria para tratar un número impar de semanas. Estas dos últimas partes han recibido mucha más atención que la primera: 28 grupos (todos salvo G05, G10, G27, G29, G30, G32) frente a 15 grupos (G01, G04, G06, G09, G10, G11, G17, G18, G19, G21, G26, G27, G29, G31, G32).

Los algoritmos incluidos en esta categoría varían en distintos detalles. Se han copiado los algoritmos tal y como los escribieron los alumnos, salvo que se han suprimido los comentarios por brevedad y algunos cambios menores realizados únicamente para que quepan en el ancho del informe.

- No se contempla la posibilidad de seleccionar un trabajo de alta tensión el primer mes. Por ejemplo (G09):

```
public static int trabajosH1(int[] a, int[] b) {
    int sumatorio = 0;
    for (int i = 0; i < a.length; i = i + 2) {
        if (i + 1 >= a.length) {
            sumatorio += b[i];
        } else if (b[i] + b[i + 1] >= a[i + 1]) {
            sumatorio += b[i] + b[i + 1];
        } else {
            sumatorio += a[i + 1];
        }
    }
    return sumatorio;
}
```

- Control del índice  $i$ . Es frecuente que la cabecera del bucle incremente el índice de dos en dos (como en los dos algoritmos anteriores), pero también existen otras variantes, como incremento de uno en uno, pero con un incremento explícito en el cuerpo del bucle (por ejemplo, G08, lo aportan incompleto):

```
public static int trabajosH1(int[] a, int[] b) {
    int i;
    int sol = 0;
    for(i=0;i<a.length;i++) {
        if(i<(a.length-1)) {
            if(b[i]+b[i+1]>a[i+1]) {
                sol += b[i]+ b[i+1];
            }else {
                sol += a[i+1];
            }
        }else {
            sol += b[i];
        }
        i++;
    }
}
```

incrementos de uno más el uso de variables auxiliares para saber si se debe descansar o no (por ejemplo, G07):

```
public static int trabajosH2 (int [] a, int [] b){
    int maxSemana;
    int beneficios=0;
    boolean descansado=true;
    for (int i=0; i<a.length;i++) {
        if(a[i]>b[i] && descansado) {
            maxSemana=a[i];
        }else {
            maxSemana=b[i];
        }
        if(i=a.length-1) {
            beneficios=beneficios+maxSemana;
        }else {
            if(maxSemana+b[i+1]>a[i+1]) {
                beneficios=beneficios+maxSemana;
                descansado=false;
            }else {
                descansado=true;
            }
        }
    }
    return beneficios;
}
```

o incrementos variables de uno o dos, según el caso (por ejemplo, G12):



```

public static int trabajosH1 (int [] a, int [] b){
    int beneficio = 0;
    for (int i=0;i<a.length;i++) {
        if (i = a.length-1){
            beneficio = beneficio + b[i];
        }
        else if (b[i]+b[i+1] > a[i+1]){
            beneficio = beneficio + b[i];
        }
        else {
            beneficio = beneficio + a[i+1];
            i++;
        }
    }
    return beneficio;
}

```

En otro caso (G19) el recorrido se realiza de forma inversa, lo que tiene la ventaja de que el primer mes no debe tratarse de forma especial:

```

public static int trabajosH3 (int[] a, int[] b){
    int i = a.length - 1;
    int beneficio = 0;
    while (i >= 0){
        if (b[i] >= a[i]){
            beneficio = beneficio + b[i];
            i--;
        }
        else{
            beneficio = beneficio + a[i];
            i = i - 2;
        }
    }
    return beneficio;
}

```

Algunos algoritmos parecen estar mal (no respetan las restricciones del problema o no suman algunos beneficios), por lo que no analizamos sus variantes (G02, G05, G20, G27, G30).

- Comparaciones distintas. Algunos grupos realizan comparaciones entre tres posibles ingresos (G23, G24, G26), lo cual permite incluir la singularidad del primer mes. Por ejemplo, G26 compara  $b[i]+b[i+1]$ ,  $a[i]+b[i+1]$  y  $a[i+1]$ :

```

public static int trabajosH2(int []a, int []b) {
    int suma=0;
    int descanso=0;
    int i=0;
    int valor1, valor2, valor3;
    while (i<a.length-1) {
        valor1= b[i]+b[i+1];
        valor2= a[i]+b[i+1];
        valor3= a[i+1];
    }
}

```

```

        if ((valor2>valor1) && (valor2>valor3) &&
            descanso==0) {
            descanso=1;
            suma += valor2;
            i=i+2;
        }else if ((valor3>valor1)) {
            descanso=1;
            suma += valor3;
            i=i+2;
        }else{
            descanso = 1;
            suma += valor1;
            i=i+2;
        }
    }
    return suma;
}

```

En tres casos (G04, G15, G24) las comparaciones se amplían a tres semanas, con la consiguiente modificación del control del bucle. Por ejemplo, G15 realiza un control variable del número de semanas:

```

public static int trabajosH3 (int[] a, int[] b) {
    int maxNumBeneficios = 0;
    for (int i = 0; i < a.length; i++) {
        int semanaInicial = i;
        int semanaFinal = i + 1;
        if (semanaInicial == 0) {
            if (a[semanaInicial] >= b[semanaInicial] &&
                a[semanaInicial] > a[semanaFinal]) {
                maxNumBeneficios += a[semanaInicial];
                continue;
            }
        }
        if (semanaInicial == a.length-1)
            maxNumBeneficios += b[semanaInicial];
        else {
            if (semanaInicial < a.length-2) {
                if (a[semanaFinal] <
                    b[semanaInicial]+a[semanaFinal+1]) {
                    maxNumBeneficios += b[semanaInicial]
                        + a[semanaFinal+1];
                    i += 2;
                }
            }
            if ((b[semanaInicial] + b[semanaFinal])
                >= a[semanaFinal])
                maxNumBeneficios += b[semanaInicial];
            else {
                maxNumBeneficios += a[semanaFinal];
            }
        }
    }
}

```

```

        i++;
    }
}
}
return maxNumBeneficios;
}

```

Por último, en otro caso (G19, incluido en la variante de control del índice  $i$ ) solamente se comparan  $a[i]$  y  $b[i]$ .

### 3.3.2 Algoritmos con Recorrido Lineal

Estos algoritmos realizan un recorrido lineal de los vectores, pero usando comparaciones distintas de las sugeridas en el enunciado de la práctica. En general, realizan menos comprobaciones que el algoritmo heurístico esbozado en el enunciado de las prácticas. En algunos casos, el beneficio se calcula sólo a partir de trabajos de alta o baja tensión, mientras que en otros casos se descartan algunas semanas o se toman los trabajos de una manera prefijada.

Por ejemplo, G04 calcula el beneficio proporcionado independientemente realizando las tareas de baja tensión en todas las semanas, las de alta tensión en semanas pares y las de alta tensión en semanas impares, eligiendo finalmente el beneficio máximo de las tres posibilidades:

```

public static int trabajosH3(int[] a, int[] b) {
    int resultadosPares = 0;
    int resultadosImpares = 0;
    int resultadosBajos = 0;
    if (a.length == 1) {
        if (a[0] >= b[0]) {
            return a[0];
        } else
            return b[0];
    }
    for (int i = 0; i < a.length; i++) {
        if (i % 2 == 0) {
            resultadosPares += a[i];
        } else {
            resultadosImpares += a[i];
        }
        resultadosBajos += b[i];
    }
    if (resultadosPares >= resultadosImpares) {
        if (resultadosPares >= resultadosBajos) {
            return resultadosPares;
        } else {
            return resultadosBajos;
        }
    }
}

```

```

    } else if (resultadosImpares >= resultadosBajos) {
        return resultadosImpares;
    } else {
        return resultadosBajos;
    }
}

```

Asimismo, G01 siempre toma un sucesivamente un trabajo de alta tensión, un trabajo de baja tensión y un descanso, comenzando desde la primera semana:

```

public static int trabajosH2(int[] a, int[] b) {
    int beneficio=0;
    for(int i=0; i<a.length; i=i+3) {
        beneficio+=a[i];
        if(i+1<a.length) {
            beneficio+= b[i+1];
        }
    }
    return beneficio;
}

```

Encontramos las siguientes variaciones, en orden decreciente de frecuencia:

- Elegir solamente trabajos de baja tensión (G01, G05, G18, G20).
- Elegir entre todos los trabajos de baja tensión, los de alta tensión en semanas pares y los de alta tensión en semanas impares (G04, G10, G16, G33).
- Elegir solamente trabajos de alta tensión en semanas pares (G01, G17, G18).
- Elegir entre trabajos de baja o de alta tensión (G01, G03, G13). G01 sólo toma trabajos en semanas pares, incluso para los de baja tensión. G13 presenta la peculiaridad de que sólo cuenta las tareas de alta tensión que son mayores que los respectivos trabajos de baja tensión.
- Elegir entre trabajos de alta tensión en semanas pares y de alta tensión en semanas impares (G05, G31).
- Elegir sucesivamente un trabajo de alta tensión, un trabajo de baja tensión y un descanso, comenzando desde la primera semana (G01, G06).
- Elegir entre trabajar sólo días pares o sólo días impares, eligiendo siempre el trabajo de mayor beneficio en la semana (G09).

### 3.2.4 Algoritmos con una Ordenación Inicial

Estos algoritmos realizan una ordenación inicial de los trabajos, como los algoritmos voraces. Solamente dos grupos (G12, G32) han usado este enfoque.

G12 ordena el vector de trabajos de alta tensión para escoger los trabajos en orden decreciente de beneficio (aunque el enunciado dice que no siempre son mayores los precios de los trabajos de alta tensión). También comprueba, para tomar un trabajo de alta tensión, que la semana anterior es de descanso o, para tomar un trabajo de baja tensión, que la siguiente está libre (es decir, no es de alta tensión, aunque el riesgo es que deje libre la actual, siendo de baja tensión la siguiente). Utiliza los vectores

auxiliares *valido* y *aAux* para guardar, respectivamente, las semanas sobre las que aún no se ha tomado ninguna decisión y el vector de indexación indirecta:

```
public static int trabajosH2(int[]a,int[]b) {
    boolean[] valido = new Boolean[a.length];
    int beneficio = 0;
    for (int i=0; i<valido.length; i++) {
        valido[i] = true;
    }
    int[] aAux = ordenarIndices(a);
    for (int i2=0; i2<a.length; i2++) {
        if (aAux[i2]==0 && valido[aAux[i2]+1]) {
            beneficio += a[aAux[i2]];
            valido[0] = false;
        } else if (aAux[i2]!=0 && aAux[i2]!=a.length-1){
            if (valido[aAux[i2]-1]) {
                beneficio += a[aAux[i2]];
                valido[aAux[i2]] = false;
            } else if (valido[aAux[i2]-1]==false
                && valido[aAux[i2]+1]) {
                beneficio += b[aAux[i2]];
                valido[aAux[i2]] = false;
            }
        } else if (aAux[i2]==a.length-1) {
            if (valido[aAux[i2]-1]==false) {
                beneficio += b[aAux[i2]];
                valido[aAux[i2]] = false;
            } else {
                beneficio += b[aAux[i2]];
                valido[aAux[i2]] = false;
            }
        }
    }
    return beneficio;
}
```

G32 también ordena las tareas de alta y de baja tensión. Intenta colocar primero las de alta tensión y después las de baja tensión, en un proceso iterativo anidado.

### 3.3 Resultados de Flexibilidad

Revisando la clasificación de los algoritmos en las tres categorías anteriores (véase Tabla 1), es fácil comprobar que sólo hay tres combinaciones de categorías por grupo. Los resultados se muestran en la Tabla 3.

**Tabla 3.** Resultados de flexibilidad en la práctica 2 (N=34)

<b>Combinaciones de categorías</b>	<b>Grupos</b>	<b># grupos</b>	<b>% grupos</b>
Enunciado	G02, G07, G08, G11, G14, G15, G19, G21, G22, G23, G24, G25, G26, G27, G28, G29, G30, G34	18	53%
Enunciado+lineal	G01, G03, G04, G05, G06, G09, G10, G13, G16, G17, G18, G20, G31, G33	14	41%
Enunciado+ordenación	G12, G32	2	6%

En definitiva, poco más de la mitad de los grupos (53%) no han tenido flexibilidad suficiente como para proponer un algoritmo heurístico con un planteamiento diferente del enunciado. En otras palabras, casi la mitad de los grupos (47%) han propuesto al menos un algoritmo con otro enfoque.

### 3.4 Resultados de Originalidad

Si consideramos las tres categorías resultantes del análisis cualitativo, es evidente que los algoritmos basados en el enunciado son poco originales, los lineales son más originales y los basados en ordenación, los más originales (aunque la ordenación se utiliza en numerosos algoritmos voraces, no suele encontrarse en los algoritmos heurísticos). A partir de esta consideración, hemos asignado valores de originalidad 1, 2 y 3 a cada algoritmo.

### 3.5 Resultados Globales de Creatividad

Medimos la creatividad de los grupos de prácticas en función de su fluidez y originalidad. Una medida sencilla consiste en sumar la originalidad de cada algoritmo. La flexibilidad de cada grupo es igual al número de categorías en las que sus algoritmos se catalogan. El enunciado de la práctica pedía desarrollar y comparar experimentalmente todos los algoritmos; por tanto, todos tienen un grado máximo de elaboración, no habiéndose tenido en cuenta este componente de la creatividad.

La Tabla 4 muestra los resultados globales. En la columna de flexibilidad, se detalla la distribución de los algoritmos desarrollados en categorías mediante una terna  $x$ - $y$ - $z$ , donde  $x$  es igual al número de algoritmos de la categoría primera (la menos original),  $y$  es el número de algoritmos de la segunda categorías y  $z$  de la tercera. La medida global de creatividad dada en la columna primera proviene de la suma ponderada de la originalidad de cada algoritmo, mediante la fórmula  $1 \cdot x + 2 \cdot y + 3 \cdot z$  a partir de la terna anteriormente citada.

**Tabla 4.** Resultados globales de creatividad en la práctica 2

<b>Creatividad global</b>	<b>Fluidez</b>	<b>Originalidad</b>	<b>Flexibilidad</b>	<b>Grupos</b>	<b># grupos</b>	<b>% grupos</b>
14	5	3-2-0 = 7	2	G01	1	3%
12	4	2-2-0 = 6	2	G18	1	3%
10	4	3-1-0 = 5	2	G04	2	6%
	3	1-2-0 = 5	2	G05		
8	3	2-1-0 = 4	2	G06, G09, G17, G20, G31	7	21%
	2	1-0-1 = 4	2	G12, G32		
6	2	1-1-0 = 3	2	G03, G10, G13, G16, G33	5	15%
4	4	4-0-0 = 4	1	G19	1	3%
3	3	3-0-0 = 3	1	G15, G22, G24, G26, G27	5	15%
2	2	2-0-0 = 2	1	G02, G07, G08, G11, G14, G21, G23, G25, G28, G29, G30, G34	12	35%

Puede observarse que aproximadamente un tercio de los grupos mostraron la mínima creatividad (con medida global igual a 2). En este caso, realizaron el mínimo esfuerzo en cuanto al número de algoritmos propios y en su originalidad.

Podemos situar en el extremo opuesto a los grupos con mayor creatividad (igual o superior a 10), apenas el 12%. Estos grupos siempre aportaron al menos un algoritmo de otra categoría (la segunda) y al menos un algoritmo más del número pedido.

### 3.6 Debate

Los resultados calculados de creatividad muestran un 12% de grupos más creativos y un tercio de grupos poco creativos. Muestran una creatividad más bien baja, en contraste con la asociación que se establece a veces entre el pensamiento algorítmico y la creatividad.

Desde un punto de vista metodológico, hemos calculado la creatividad global de cada grupo de una manera particular, que conviene resaltar:

- La creatividad de cada grupo se ha calculado como: (a) la suma de la originalidad de los algoritmos desarrollados, y (b) la multiplicación de dicho valor por la flexibilidad.
- Se ha realizado un análisis cualitativo de los algoritmos heurísticos que permite medir su originalidad y flexibilidad.
- La originalidad se ha calculado en función de su semejanza con una solución esbozada por el profesor.
- No se ha tenido en cuenta la elaboración de cada algoritmo ya que, al haberse programado todos, se ha considerado que todos tenían la misma elaboración. Evidentemente, podrían tenerse en cuenta otros factores (explicación, comentarios) para tomar una decisión distinta sobre su elaboración.
- La flexibilidad se ha medido como el número de categorías en las que se clasifican las aportaciones realizadas de algoritmos heurísticos.

## **4 Resultados de la Práctica 3b – Ramificación y Poda**

El enunciado de la práctica 3b pedía una o más funciones de cota para el problema de planificación de trabajos con beneficio máximo. Dado que es un problema de maximización, cada función debía calcular una cota superior del beneficio optimal.

A continuación, se presentan sucesivamente las entregas de los alumnos, los resultados de fluidez, un análisis cualitativo de las funciones de cota, los resultados de flexibilidad, originalidad y elaboración, y la creatividad global.

### **4.1 Entregas de los Alumnos**

Se recogieron 31 informes de prácticas, 2 del grupo online y 29 del grupo presencial. De ellas, 27 correspondían a parejas de alumnos (1 del grupo online y 26 del grupo presencial) y 4 a alumnos individuales (1 del grupo online y 3 del grupo presencial).

En la Tabla 5 incluimos las entregas recibidas. Se presenta el número de cada grupo de prácticas, si era una pareja o un alumno solo, el número de entrega, una breve descripción de las características de cada función de cota propuesta, la categoría de cada aportación, si se ha programado y si garantiza una cota superior.

Obsérvese que se permitió realizar una segunda entrega. Por sencillez, la Tabla 5 solamente recoge los 6 envíos en los que varió alguna función de cota, contabilizando un total de 37 envíos. Sin embargo, las segundas entregas no se analizan en los apartados siguientes, ya que se realizaron como respuesta a los comentarios del profesor.



**Tabla 5.** Informes de prácticas presentados para la práctica 3b

Nº grupo	Grupo de clase	Individual /pareja	Entrega	Funciones de cota	Programado	Cota superior
1.	OL	I	1	1. Suma de trabajos de alta tensión 2. Máximo trabajo de baja tensión por el número de semanas más una 3. Media de trabajos de baja y alta tensión por el número de semanas más dos	1. Sí 2. Sí 3. Sí	1. Sí* 2. No** 3. No**
2.	OL	P	1	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí
			2	-		
3.	Pr	P	1	1. Suma de trabajos de alta tensión	1. Sí	1. Sí*
4.	Pr	P	1	1. Suma de trabajos de alta tensión	1. Sí	1. Sí*
				2. Suma de todos los beneficios de baja y alta tensión	2. Sí	2. Sí
5.	Pr	P	1	1. Suma de todos los beneficios de baja y alta tensión	1. Sí	1. Sí
6.	Pr	P	1	1. Suma de trabajos de baja tensión	1. Sí	1. No
			2	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí
7.	Pr	P	1	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí
8.	Pr	P	1	1. Suma de trabajos de baja tensión	1. Sí	1. No
			2	-		
9.	Pr	P	1	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí
10.	Pr	P	1	1. Suma de todos los beneficios de baja y alta tensión	1. Sí	1. Sí
11.	Pr	I	1	1. Suma de todos los beneficios de baja y alta tensión	1. Sí	1. Sí
12.	Pr	P	1	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí
13.	Pr	P	1	1. Coste de la solución parcial (técnica de vuelta atrás)	1. Sí	1. No
			2	1. Suma de todos los beneficios de baja y alta tensión	1. Sí	1. Sí
14.	Pr	P	1	1. Coste de la solución parcial (técnica de vuelta atrás)	1. Sí	1. No
			2	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí
15.	Pr	P	1	1. Suma de trabajos de alta tensión	1. Sí	1. Sí*
16.	Pr	P	1	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí
17.	Pr	P	1	1. Coste de la solución parcial (técnica de vuelta atrás)	1. Sí	1. No
			2	1. Heurística	1. Sí	1. No

18.	*Pr	I	1	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí
			2	-		
19.	Pr	P	1	1. Suma de todos los beneficios de baja y alta tensión 2. Suma del máximo trabajo de cada semana 3. Heurística	1. No 2. Sí 3. No	1. Sí 2. Sí 3. No
20.	Pr	P	1	1. Valor máximo hasta el momento (técnica de vuelta atrás)	1. Sí	1. No
21.	Pr	P	1	1. Suma de trabajos de alta tensión 2. Suma de trabajos de baja tensión 3. Suma del máximo trabajo de cada semana	1. No 2. No 3. Sí	1. Sí* 2. No 3. Sí
22.	Pr	P	1	1. Comprobación adicional de validez (técnica de vuelta atrás) 2. Suma del máximo trabajo de cada semana 3. Suma del máximo trabajo de cada semana y comprobación de validez	1. Sí (VA2) 2. Sí (VA3) 3. Sí (VA4)	1. No 2. Sí 3. Sí
23.	Pr	P	1	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí
24.	Pr	P	1	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí
25.	Pr	P	1	1. Suma del máximo trabajo de cada semana 2. Heurística	1. Sí 2. Sí	1. Sí 2. No
26.	Pr	P	1	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí
27.	Pr	P	1	1. Suma de trabajos de alta tensión	1. Sí	1. Sí*
28.	Pr	P	1	1. Comprobación adicional de validez (técnica de vuelta atrás)	1. Sí	1. No
			2	1. Máximo beneficio en la primera semana más suma de trabajos de baja tensión para el resto	1. Sí	1. No
29.	Pr	P	1	1. Heurística	1. Sí	1. No
			2	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí
30.	Pr	I	1	1. Suma de trabajos de alta tensión	1. Sí	1. Sí*
31.	Pr	P	1	1. Suma del máximo trabajo de cada semana	1. Sí	1. Sí

\*: Bien, suponiendo que todos los trabajos de alta tensión se pagan mejor que los de baja tensión.

\*\* : El propio grupo es consciente de que no es una cota superior. Intenta arreglarlo multiplicando por un número mayor de semanas.

## 4.2 Resultados de Fluidez

La Tabla 6 muestra los resultados de fluidez, mostrando el número y porcentaje de grupos que han presentado funciones de cota.

**Tabla 6.** Resultados de fluidez en la práctica 3b (N=31)

Número de funciones de cota	Grupos	# grupos	% grupos
1	G02, G03, G05, G06, G07, G08, G09, G10, G11, G12, G13, G14, G15, G16, G17, G18, G20, G23, G24, G26, G27, G28, G29, G30, G31	25	82%
2	G04, G25	2	6%
3	G01, G19, G21, G22	4	12%

Los grupos que han realizado más de una propuesta suman un 18%.

## 4.3 Categorías de Algoritmos

En este caso, resulta relativamente sencillo establecer categorías de las aportaciones, según las características de la cota asociada:

- Cota superior. Garantizan una cota superior sobre el beneficio optimal. A su vez, podemos distinguir las siguientes cotas:
  - Suma del máximo trabajo de cada semana (G02, G07, G09, G12, G16, G18, G19.2, G21.3, G22.2, G23, G24, G25.1, G26, G31).
  - Suma de todos los beneficios de baja y alta tensión (G04.2, G05, G10, G11, G19.1).
- Cota superior posible. Pueden garantizar una cota superior sobre el beneficio optimal bajo ciertas condiciones (p.ej. que los beneficios de los trabajos de alta tensión siempre sean mayores que los de baja tensión). A su vez, podemos distinguir las siguientes cotas:
  - Suma de trabajos de alta tensión. Sería una cota superior si los beneficios generados para trabajos de alta tensión siempre son mayores que los de baja tensión (G01.1, G03, G04.1, G15, G21.1, G27, G30).
  - Cierta beneficio multiplicado por más semanas de las que hay de trabajo (G01.2, G01.3).
- Cota inferior. Proporcionan una cota inferior sobre el beneficio optimal. A su vez, podemos distinguir las siguientes cotas:
  - Suma de trabajos de baja tensión (G06, G08, G21.2).
  - Valor calculado con una heurística (G19.3, G25.2, G29).
- No son función de cota. Se destacan o modifican distintos elementos ya presentes en la técnica de vuelta atrás:
  - Coste de la solución parcial (G13, G14, G17).
  - Máximo valor calculado hasta el momento (G20).

- Comprobación adicional de validez (G22.1, G22.3, G28).

Cabe mencionar que el grupo G28 realizó una segunda entrega con una cota inferior: el máximo beneficio en la primera semana más la suma de trabajos de baja tensión para el resto.

#### 4.4 Resultados de Flexibilidad

Consideramos que las oportunidades de flexibilidad son limitadas. Los grupos que han aportado varias funciones de cota ya han mostrado fluidez. No tiene sentido considerar flexibilidad cuando se hacen propuestas que corresponden a cotas inferiores o que, incluso, no son funciones de cota. Es posible que hayan realizado estas propuestas porque no se les ocurrían otras. Por tanto, hemos considerado que muestran flexibilidad aquellos grupos que proponen distintas cotas superiores o, en todo caso, los que proponen posibles cotas superiores.

Los resultados se muestran en la Tabla 7.

**Tabla 7.** Resultados de flexibilidad en la práctica 3b (N=31)

<b>Grado de flexibilidad</b>	<b>Grupos</b>	<b># grupos</b>	<b>% grupos</b>
Flexibilidad	G01, G19	2	87%
Posible flexibilidad	G04, G21	2	6%
Inflexibilidad	G02, G03, G05, G06, G07, G08, G09, G10, G11, G12, G13, G14, G15, G16, G17, G18, G20, G22, G23, G24, G25, G26, G27, G28, G29, G30, G31	27	6%

En definitiva, solamente el 12% de los grupos han mostrado flexibilidad o “posible flexibilidad”.

#### 4.5 Resultados de Originalidad y Elaboración

El enunciado de la práctica 3b no da ninguna indicación de cómo calcular una cota; ni siquiera, que deba ser una cota superior. Por tanto, parece razonable que midamos la originalidad de cada función de cota como por frecuencia de contribución. En el caso de propuestas que no corresponden a cotas, podemos agruparlas en una única clase, con originalidad cero.

La Tabla 8 muestra cada función de cota. La penúltima columna contiene el número de grupos que la proponen y su porcentaje correspondiente. La última columna contiene la originalidad de cada función de cota. Se ha calculado a partir de la inversa del porcentaje de algoritmos que son iguales, de forma que aquellos algoritmos con mayor porcentaje, tienen una inversa menor. La inversa de cada algoritmo se ha normalizado al intervalo [0..1], para que todos los criterios de originalidad tengan el mismo peso. La categoría que agrupa las propuestas que no son cotas no se tiene en cuenta para la originalidad, asignándoles una originalidad nula.

**Tabla 8.** Resultados de originalidad de cada función de cota (N=34)

<b>Función de cota</b>	<b>Aportaciones</b>	<b># (%)</b>	<b>1 / % norm.</b>
Suma del máximo trabajo de cada semana	G02, G07, G09, G12, G16, G18, G19.2, G21.3, G22.2, G23, G24, G25.1, G26, G31	14 (%)	0,21
Suma de todos los beneficios de baja y alta tensión	G04.2, G05, G10, G11, G19.1	5 (%)	0,4
Suma de trabajos de alta tensión	G01.1, G03, G04.1, G15, G21.1, G27, G30	7 (%)	0,29
Cierto beneficio multiplicado por más semanas	G01.2, G01.3	2 (%)	1
Suma de trabajos de baja tensión	G06, G08, G21.2	3 (%)	0,67
Valor calculado con una heurística	G19.3, G25.2, G29	3 (%)	0,67
No son función de cota	G13, G14, G17, G20, G22.1, G22.3, G28	–	0

La Tabla 9 recoge la originalidad de cada función de cota, así como su grado de elaboración, según el siguiente criterio. Las funciones de cota codificadas en algoritmos de ramificación y poda tienen una elaboración igual a la unidad, y aquellos que solamente se describen, obtienen una elaboración de dos tercios. Obsérvese que los alumnos no habrían necesitado mucha más elaboración para codificarlas, ya que simplemente tenían que modificar la codificación de otro algoritmo de ramificación y poda. La última columna contiene el producto de la originalidad de cada función de cota por su grado de elaboración, producto que se utilizará en el próximo apartado para calcular la creatividad global de cada grupo.

**Tabla 9.** Resultados de originalidad por función de cota (N=41)

<b>Función de cota</b>	<b>Originalidad</b>	<b>Elaboración</b>	<b>Producto</b>
G01.1	0,29	1	0,29
G01.2	1	1	1
G01.3	1	1	1
G02	0,21	1	0,21
G03	0,29	1	0,29
G04.1	0,29	1	0,29
G04.2	0,4	1	0,4
G05	0,4	1	0,4
G06	0,67	1	0,67
G07	0,21	1	0,21
G08	0,67	1	0,67
G09	0,21	1	0,21
G10	0,4	1	0,4
G11	0,4	1	0,4
G12	0,21	1	0,21
G13	0	1	0
G14	0	1	0
G15	0,29	1	0,29
G16	0,21	1	0,21

G17	0	1	0
G18	0,21	1	0,21
G19.1	0,4	0,67	0,27
G19.2	0,21	0,67	0,14
G19.3	0,67	1	0,67
G20	0	1	0
G21.1	0,29	0,67	0,19
G21.2	0,67	0,67	0,45
G21.3	0,21	1	0,21
G22.1	0	1	0
G22.2	0,21	1	0,21
G22.3	0	1	0
G23	0,21	1	0,21
G24	0,21	1	0,21
G25.1	0,21	1	0,21
G25.2	0,67	1	0,67
G26	0,21	1	0,21
G27	0,29	1	0,29
G28	0	1	0
G29	0,67	1	0,67
G30	0,29	1	0,29
G31	0,21	1	0,21

#### 4.6 Resultados Globales de Creatividad

Medimos la creatividad de los grupos de prácticas en función de su fluidez, flexibilidad, originalidad y elaboración. La Tabla 10 contiene, para cada grupo, su originalidad global, su fluidez, la suma del producto originalidad-elaboración obtenido para cada una de sus funciones de cota y su flexibilidad. La creatividad calculada para cada grupo es igual al producto de estas dos últimas columnas. Finalmente, se da el número y porcentaje de grupos asociados a cada valor global resultante de originalidad.

**Tabla 10.** Resultados globales de creatividad en la práctica 3b

Grupo	Creatividad global	Fluidez	Suma de productos OE	Flexibilidad	# grupos	% grupos
G01	4,58	3	2,29	2	1	3%
G19	2,16	3	1,08	2	1	3%
G21	1,275	3	0,85	1,5	1	3%
G04	1,035	2	0,69	1,5	1	3%
G25	0,88	2	0,88	1	1	3%
G06	0,67	1	0,67	1	3	10%
G08						
G29						
G05	0,4	1	0,4	1	3	10%
G10						
G11						

G03	0,29	1	0,29	1	4	13%
G15						
G27						
G30						
G02	0,21	1	0,21	1	10	32%
G07						
G09						
G12						
G18						
G22						
G23						
G24						
G26	0,15	1	0,15	1	1	3%
G31						
G16	0	1	0	1	5	16%
G13						
G14						
G17						
G20						
G28						

Puede observarse que hay cuatro bloques de grupos, según su originalidad global. En el extremo superior, hay cuatro grupos (12%) con un grado de originalidad mayor que la unidad. Todos los grupos han mostrado fluidez y flexibilidad. Han presentado cotas de distintas clases (cota superior, posible cota superior o cota inferior).

A continuación, tenemos otros cuatro grupos (12%) con una flexibilidad media, con un grado entre medio punto y la unidad. Sin embargo, todos ellos propusieron cotas inferiores.

El tercer bloque engloba 18 grupos (58%) que presentan un grado de creatividad inferior a medio punto. Han presentado una cota superior o posible cota superior.

El último bloque corresponde a cinco grupos (16%) que hicieron una aportación que no correspondía a una cota.

#### 4.7 Debate

Los resultados calculados de creatividad vuelven a mostrar un porcentaje reducido (12%) de grupos más creativos y un porcentaje alto de grupos poco creativos (58%) o con contribuciones gravemente erróneas (16%).

Desde un punto de vista metodológico, hemos calculado la creatividad global de cada grupo de una manera particular, que conviene resaltar:

- La originalidad de cada función de cota se midió en función de su frecuencia porque el enunciado de la práctica no daba ninguna pista sobre su diseño, ni siquiera si debía ser una cota inferior o superior.
- No se ha querido penalizar excesivamente la originalidad de las “posibles” cotas superiores, aunque los alumnos ya habían sido advertidos de la práctica 2 que los trabajos de alta tensión no tenían por qué proporcionar siempre un beneficio mayor que las de baja tensión.

- Se han tomado dos decisiones que redundan en una creatividad nula:
  - Originalidad de aquellas aportaciones que no son cotas, sino elementos de la técnica de vuelta atrás. Consideramos que no tiene sentido hablar de su creatividad.
  - Flexibilidad de aquellas aportaciones que son cotas inferiores o que son elementos de la técnica de vuelta atrás. Consideramos que son aportaciones realizadas sin apenas fundamento.
- La creatividad de cada grupo se ha calculado en varios pasos:
  - La originalidad de cada función de cota es igual a la inversa normalizada de su frecuencia.
  - La originalidad de cada función de cota se multiplicó por su elaboración, según que se hubiera programado o no.
  - Para cada grupo, se sumó el valor originalidad-elaboración de cada función de cota y el resultado se multiplicó por su flexibilidad.

Por último, hay que destacar que la medida de la creatividad no tiene por qué corresponder con la calidad de las aportaciones. Se ha visto que los grupos con mayor originalidad presentan cotas de todo tipo. Asimismo, el bloque de grupos con originalidad media presenta soluciones peores (cotas inferiores) que el bloque con creatividad baja (cotas superiores o posibles cotas superiores).

## 5 Resultados de la Práctica 6 – Algoritmos Probabilistas

A continuación, se presentan sucesivamente las entregas de los alumnos, los resultados de fluidez, un análisis cualitativo de los algoritmos probabilistas, los resultados de flexibilidad, originalidad y elaboración, y la creatividad global de cada grupo.

### 5.1 Entregas de los Alumnos

Se recogieron 11 informes de prácticas del grupo presencial, 8 correspondientes a parejas de alumnos y 3 a alumnos individuales. Al ser la última práctica y no ser obligatoria, se recogieron pocas prácticas y ninguna de ellas en una segunda entrega.

En la Tabla 11 incluimos las entregas recibidas. Se presenta el número de cada grupo de prácticas, si era una pareja o un alumno solo, una breve descripción de las características del algoritmo probabilista, y tres criterios que caracterizan los algoritmos presentados: enfoque principal del algoritmo, rango de la generación de números aleatorios y número de generaciones de números aleatorios efectuadas en cada ocasión.



**Tabla 11.** Informes de prácticas presentados para la práctica 6

Nº grupo	Indiv./ pareja	Algoritmos	Enfoque	Rango de aleatoriedad	Núm. intentos aleatorios
G04	P	Para cada objeto, itera aleatoriamente hasta que logra introducirlo en alguna caja.	Inseguro	Cajas creadas	Los necesarios
G08	P	Aleatoriamente, intenta meterlo en la primera caja y en alguna otra. En el segundo caso, recorre las cajas secuencialmente y la mete en la primera que pueda, salvo que haya que crear una,	Inseguro	La primera o la primera creada donde quepa	Uno
G12	I	Para cada objeto, itera aleatoriamente en las cajas disponibles hasta que logra introducirlo en alguna caja (si no cabe en ninguna, toma una nueva).	Inseguro	Cajas creadas	Los necesarios
G15	P	Para cada objeto, se elige aleatoriamente entre las cajas con espacio suficiente y crear una nueva.	Inseguro	Cajas con espacio	Uno
G16	P	Para cada objeto, se elige aleatoriamente entre las cajas con espacio suficiente y crear una nueva.	Inseguro	Cajas con espacio	Uno
G18	I	1. Para cada objeto, elige aleatoriamente y dependiendo de su peso si meterlo en una caja nueva o en la primera que tenga espacio libre. 2. Hace 3 variantes, disminuyendo la probabilidad de que el objeto se introduzca en una caja nueva.	1. Inseguro 2. Inseguro	1. Nueva o la primera creada donde quepa 2. Nueva o la primera creada donde quepa	1. Uno 2. Uno
G19	P	1. Para cada objeto, se elige aleatoriamente entre meterla en una caja creada o crear una caja nueva 2. Para cada objeto, se elige aleatoriamente entre meterla en una caja creada o crear una caja nueva	1. Inseguro 2. Inseguro	1. Cajas creadas 2. Cajas creadas	1. Uno 2. Los necesarios
G23	P	1. Elige aleatoriamente una caja; si no cabe, elige aleatoriamente otra; sin no cabe, se mete en una caja nueva. 2. Sugieren otro algoritmo, consistente en que la elección aleatoria no sea de las cajas ocupadas temporalmente, sino de la cota inferior de cajas previstas.	Inseguro	1. Cajas creadas 2. Previsión de cajas creadas	1. Dos 2. Dos
G24	P	Para cada objeto, itera aleatoriamente hasta que logra introducirlo en alguna caja.	Inseguro	Cajas creadas no llenas	Los necesarios
G29	P	1. Usa un algoritmo mixto pero inseguro, que combina la colocación aleatoria de un cuarto de los objetos en la primera caja donde quepa, más la colocación mediante vuelta atrás del resto. 2. Desarrolla un algoritmo seguro, comparando sus resultados con una cota inferior el número de cajas.	1. Mixto inseg. 2. Mixto seguro	1. Todos los objetos 2. Todos los objetos	1. Uno 2. Uno
G30	I	Para cada objeto, itera aleatoriamente hasta que logra introducirlo en alguna caja.	Inseguro	Cajas creadas	Los necesarios

## 5.2 Resultados de Fluidez

La Tabla 12 muestra los resultados de fluidez, mostrando el número y porcentaje de grupos que han presentado diversos números de algoritmos.

**Tabla 12.** Resultados de fluidez en la práctica 6 (N=11)

Número de algoritmos	Grupos	# grupos	% grupos
1	G04, G08, G12, G15, G16, G23, G24, G30	7	64%
2	G19, G23, G29	3	27%
4	G18	1	9%

Los grupos han propuesto un solo algoritmo o, en todo caso, una variante. El grupo G18 es un caso especial porque presenta 3 variantes del mismo algoritmo, que sólo varían respecto al algoritmo original en un porcentaje que va probando empíricamente.

## 5.3 Categorías de Algoritmos

Resulta difícil establecer categorías de algoritmos. Resulta más fácil distinguir clases de algoritmos según los tres criterios incluidos en las últimas tres columnas de la Tabla 11:

- Enfoque. Este criterio puede tomar los valores inseguro, mixto inseguro y mixto seguro. Todos los grupos tienen un enfoque inseguro, salvo G29 que utiliza los dos enfoques mixtos. No obstante, no tiene sentido el enfoque mixto seguro porque, para saber cuándo una ejecución es segura, utiliza como referencia ¡un algoritmo de vuelta atrás (exacto, pero más ineficiente) que resuelve el mismo problema!
- Rango de aleatoriedad. Resume cuál es el rango de índices o valores sobre los cuales se realiza la operación de generación de números aleatorios. Encontramos los siguientes casos:
  - Cajas creadas (G04, G12, G19, G23.1, G30).
  - Cajas creadas y con espacio para que quepa el objeto (G15, G16).
  - Elegir entre dos opciones: comenzar una búsqueda secuencial de una caja donde quepa el objeto y bien la primera caja creada (G08) bien una caja nueva (G18).
  - Una previsión (cota inferior) del número de cajas necesarias (G23.2).
  - Cajas creadas y no llenas (G24).
  - Objetos, para seleccionar el siguiente objeto a meter (G29).
- Número de intentos aleatorios. Detalla el número de operaciones de generación de números aleatorios que se realiza cada vez:
  - Una sola. Corresponde a varios casos: se elige una caja de aquellas donde seguro que cabe un objeto (G15, G16), se toma una decisión poco relevante por buscarse una caja secuencialmente (G08, G18), se toma una

- decisión entre caja nueva o ya existente (G19.1) o se toma una decisión poco relevante por ser un algoritmo mixto (G29).
- Dos. Es un caso peculiar, donde solamente prueba aleatoriamente con dos cajas antes de crear una nueva (G23).
  - Todas las necesarias. Normalmente se realiza cuando se intenta asignar meter un objeto en una caja de las creadas, pero no se ha comprobado si cabe (G04, G12, G19.2, G30). Como mucho, se comprueba que son cajas no llenas (G24).

#### 5.4 Resultados de Flexibilidad

Dado que sólo ha habido cuatro entregas de dos o más algoritmos, podemos considerar que han sido flexibles los grupos cuyas dos entregas han realizado una variación sustancial en alguno de los tres criterios (véase Tabla 11). Los resultados se muestran en la Tabla 13.

**Tabla 13.** Resultados de flexibilidad en la práctica 6 (N=11)

Cambio relevante en criterio	Grupos	# grupos	% grupos
Sin cambio	G04, G08, G12, G15, G16, G18, G24, G30	8	73%
Con cambio	G19, G23, G29	3	27%

En definitiva, poco más de la cuarta parte de los grupos han tenido flexibilidad suficiente como para proponer un cambio destacable en sus algoritmos. Recordemos que el grupo G18 hizo pequeños cambios en un porcentaje, pero sin variar el algoritmo.

#### 5.5 Resultados de Originalidad y Elaboración

El enunciado de la práctica 6 indicaba que se eligiera aleatoriamente una caja entre aquéllas en las cuales podía haber cada objeto (véase el Apéndice). Sin embargo, solamente dos grupos (G15, G16, véase apartado 5.3) usaron esta estrategia. Dado el poco caso prestado a nuestra sugerencia de diseño, parece más razonable que midamos la originalidad como la frecuencia de cada contribución. Dado que hemos identificado tres criterios, podemos calcular la originalidad de cada algoritmo en cada criterio y posteriormente calcular su media.

Las tres tablas siguientes presentan el resultado de calcular la originalidad de cada algoritmo probabilista según cada uno de los tres criterios: enfoque en la Tabla 14, rango de aleatoriedad en la Tabla 15 y el número de intentos aleatorios en la Tabla 16. La penúltima columna contiene el número de algoritmos y el porcentaje de cada categoría. La última columna contiene la originalidad de cada algoritmo. Se ha calculado como la inversa del porcentaje de algoritmos que son iguales, de forma que aquellos algoritmos con mayor porcentaje, su inversa es menor. La inversa de cada

algoritmo se ha normalizado finalmente al intervalo [0..1], para que todos los criterios de originalidad tengan el mismo peso.

**Tabla 14.** Resultados de originalidad según el enfoque (N=17)

Categoría	Algoritmos	# (%)	1 / % norm.
Inseguro	G04, G08, G12, G15, G16, G18.1, G18.2, G18.3, G18.4, G19.1, G19.2, G23.1, G23.2, G24, G30	15 (88%)	0,07
Mixto inseguro	G29.1	1 (6%)	1
Mixto seguro	G29.2	1 (6%)	1

**Tabla 15.** Resultados de originalidad según el rango de aleatoriedad (N=17)

Categoría	Algoritmos	# (%)	1 / % norm.
Cajas creadas	G04, G12, G19.1, G19.2, G23.1, G30	6 (35%)	0,17
Dos opciones	G08, G18.1, G18.2, G18.3, G18.4	5 (29%)	0,2
Cajas creadas con espacio	G15, G16	2 (12%)	0,5
Objetos	G29.1, G29.2	2 (12%)	0,5
Previsión de cajas	G23.2	1 (6%)	1
Cajas creadas no llenas	G24	1 (6%)	1

**Tabla 16.** Resultados de originalidad según el número de intentos aleatorios (N=17)

Categoría	Algoritmos	# (%)	1 / % norm.
Uno	G08, G15, G16, G18.1, G18.2, G18.3, G18.4, G19.1, G29.1, G29.2	10 (%)	0,2
Los necesarios	G04, G12, G19.2, G24, G30	5 (%)	0,4
Dos	G23.1, G23.2	2 (12%)	1

La Tabla 17 recoge los valores de originalidad de cada algoritmo para cada criterio, calculados en las tres tablas anteriores, así como su media.

También recoge el grado de elaboración de cada algoritmo, según el siguiente criterio. Todos los algoritmos codificados tienen una elaboración igual a la unidad, salvo el algoritmo 23.2, que se explica verbalmente pero no se desarrolla. La última columna contiene el producto de la originalidad (media) de cada algoritmo por su grado de elaboración, producto que se utilizará en el próximo apartado para calcular la creatividad global de cada grupo.

**Tabla 17.** Resultados de originalidad por algoritmo (N=17)

Algoritmo	Enfoque	Rango de aleatoriedad	Número de intentos aleatorios	Originalidad media	Elaboración	Producto
G04	0,07	0,17	0,4	0,21	1	0,21
G08	0,07	0,2	0,2	0,16	1	0,16
G12	0,07	0,17	0,4	0,21	1	0,21
G15	0,07	0,5	0,2	0,26	1	0,26
G16	0,07	0,5	0,2	0,26	1	0,26

G18.1	0,07	0,2	0,2	0,16	1	0,16
G18.2	0,07	0,2	0,2	0,16	1	0,16
G18.3	0,07	0,2	0,2	0,16	1	0,16
G18.4	0,07	0,2	0,2	0,16	1	0,16
G19.1	0,07	0,17	0,2	0,15	1	0,15
G19.2	0,07	0,17	0,4	0,21	1	0,21
G23.1	0,07	0,17	1	0,41	1	0,41
G23.2	0,07	1	1	0,69	0,5	0,35
G24	0,07	1	0,4	0,49	1	0,49
G29.1	1	0,5	0,2	0,57	1	0,57
G29.2	1	0,5	0,2	0,57	1	0,57
G30	0,07	0,17	0,4	0,21	1	0,21

## 5.6 Resultados Globales de Creatividad

Medimos la creatividad de los grupos de prácticas en función de su fluidez, flexibilidad, originalidad y elaboración. La Tabla 18 contiene, para cada grupo, su fluidez, la suma del producto de originalidad y elaboración obtenido para cada uno de sus algoritmos y su flexibilidad. La creatividad calculada para cada grupo es igual al producto de la columna tercera por su flexibilidad.

**Tabla 18.** Resultados globales de creatividad en la práctica 6

Grupo	Fluidez	Suma de productos OE	Flexibilidad	Creatividad global
G29	2	1,14	2	2,28
G23	2	0,76	2	1,52
G19	2	0,36	2	0,72
G18	4	0,64	1	0,64
G24	1	0,49	1	0,49
G15	1	0,26	1	0,26
G16				
G04	1	0,21	1	0,21
G12				
G30				
G08	1	0,16	1	0,16

Puede comprobarse que hay tres grupos de prácticas, según su creatividad. Dos grupos (18%) obtienen altos valores comparativos de creatividad. Ambos grupos aportaron dos algoritmos y cambiaron un criterio de originalidad. El grupo G29 desarrolló completamente ambos algoritmos, que mantuvieron un alto valor en dos criterios de originalidad. El grupo G23 sólo elaboró a medias su segundo algoritmo, pero quedó compensado por su originalidad según dos criterios.

Un segundo bloque formado por tres grupos de prácticas obtuvo valores medios de creatividad, entre aproximadamente medio punto y la unidad. El grupo G19 presenta flexibilidad en un criterio. El grupo G18 presenta alta fluidez, pero sin flexibilidad. Finalmente, el grupo G24 presenta un solo algoritmo con alta originalidad según un criterio y media según otro.

Finalmente, el bloque con los restantes seis grupos (55%) presenta la menor creatividad. Son grupos que solamente han aportado un algoritmo, poco original en los distintos criterios.

## 5.7 Debate

Los resultados calculados de creatividad vuelven a mostrar un porcentaje relativamente reducido (18%) de grupos más creativos y un porcentaje alto (55%) de grupos poco creativos.

Desde un punto de vista metodológico, hemos calculado la creatividad global de cada grupo de una manera particular, que conviene resaltar:

- La creatividad de cada grupo se ha calculado en varios pasos:
  - La originalidad de cada algoritmo era igual a la media de su originalidad según tres criterios. La originalidad de un algoritmo según un criterio es igual a la inversa normalizada de su frecuencia.
  - La originalidad de cada algoritmo se multiplicó por su elaboración, según que se hubiera programado o no.
  - Para cada grupo, se sumó el valor de originalidad-elaboración de cada algoritmo y el resultado se multiplicó por su flexibilidad.
- El grupo G18 plantea la cuestión de cómo medir la fluidez, aparentemente el criterio más sencillo de operacionalizar. Las variantes que pone solamente varían en un porcentaje utilizado en un solo lugar del algoritmo, así que son variantes del mismo algoritmo. Por tanto, podrían interpretarse como intentos, documentados, de elaborar un mismo algoritmo. En este caso, la elaboración de un algoritmo debería tener una definición más compleja, difícil de medir cuando se trata de una característica peculiar de un solo algoritmo.
- La originalidad de cada algoritmo se midió en función de su frecuencia en lugar de su “distancia” con el propuesto por el profesor porque su propuesta apenas fue tenida en cuenta por los alumnos.

Por último, hay que destacar que la medida de la creatividad no tiene por qué corresponder con la calidad de los algoritmos. En realidad, podemos apreciar cierta tensión entre algunos de los componentes de la creatividad:

- Normalmente, los alumnos más trabajadores o más brillantes suelen proponer más soluciones, redundando en una mayor fluidez, elaboración y, con frecuencia, en mayor flexibilidad.
- Una mayor originalidad (incluso la elaboración, arriba comentada, de G18) puede corresponder a soluciones de poca calidad o poco adecuadas para el problema planteado.

## 6 Debate

Podemos destacar los siguientes resultados obtenidos sobre la creatividad de cada práctica:

- En general, el porcentaje de grupos con valores altos de creatividad global, así como de sus componentes, es bajo, situándose entre el 10 y 20% aproximadamente. Asimismo, el porcentaje de grupos por mínima creatividad suele ser alto, por encima del 50%.
- El valor global de creatividad no siempre tiene relación con la calidad del trabajo realizado. Suele ser así en las tareas donde hay más libertad (como es el caso de los algoritmos heurísticos o probabilistas de las prácticas 2 y 6) pero en aquellas donde se exige mayor rigor (como las funciones de cota de la práctica 3b), es probable que obtengan mayores valores de creatividad los alumnos que realizan propuestas peores (quizá, poco meditadas) que alumnos con mejores soluciones (poco originales pero correctas).
- Podemos estar midiendo como creatividad otros constructos, como la actitud o habilidades y estrategias metacognitivas. De hecho, hemos observado que alumnos aparentemente poco brillantes, pero perseverantes, obtienen resultados altos de creatividad, principalmente por una mayor fluidez.

Por otro lado, desde un punto de vista metodológico, podemos extraer las siguientes conclusiones:

- La fluidez es el componente más sencillo de medir, igual al número de aportaciones distintas realizadas. Sin embargo, pueden existir casos en los que las variaciones entre aportaciones son relativamente menores. En nuestro caso, esto se hizo patente en los cuatro algoritmos presentados por G18 en la práctica 6, que más bien corresponden a distintos intentos de un mismo algoritmo para conseguir mayor grado de elaboración.
- Es necesario un análisis cualitativo que permita establecer categorías de aportaciones. Lo elaborado del análisis cualitativo depende de los objetivos de cada práctica. Hemos visto que en las prácticas 2 y 6, más abiertas, surgían diversos detalles a considerar, pero finalmente bastaba con distinguir dos o tres categorías de algoritmos. Sin embargo, en la práctica 3b, más cerrada, bastaba con distinguir la corrección de las aportaciones.
- La originalidad se ha medido de forma distinta en la práctica 2, donde se daba una orientación clara por parte del profesor, y en las otras dos prácticas, donde no se daba (o no hicieron caso). En el primer caso, se medía la “distancia” entre la categoría asignada a cada algoritmo por el análisis cualitativo y la categoría sugerida por el profesor. En el segundo caso, se midió su frecuencia, aunque también tratada de forma distinta. En la más cerrada práctica 3b se midió la frecuencia de cada contribución “aceptable”, mientras que en la más abierta práctica 6 se midió la frecuencia de varios aspectos clave y se calculó su media.
- El grado de elaboración de las aportaciones algorítmicas se ha valorado según que se presentaran programadas o no. Aunque en términos generales, ha resultado una decisión adecuada, en la práctica 3b la diferencia entre definir

una cota con detalle y codificarla era pequeña. En otros casos puede ser insuficiente y quizá podría medirse según algún criterio de calidad, como el porcentaje de ejecuciones optimales en las prácticas 2 y 6 o la eficiencia en la práctica 3b. También podría tenerse en cuenta la documentación o las explicaciones proporcionadas, aunque son criterios menos objetivos.

- La flexibilidad mide la relación entre las distintas aportaciones. Por tanto, tiene una relación directa con las categorías del análisis cualitativo. En las tres prácticas se tuvo en cuenta la variación entre categorías. Sin embargo, en la práctica 3b se dio flexibilidad y originalidad nulas a aportaciones poco razonables, que podían corresponder a “palos de ciego”.
- Hemos medido de distintas formas la originalidad global a partir de sus distintos componentes. En general, siempre hemos sumado la fluidez de la originalidad de cada aportación. En las prácticas 3b y 6, se ha matizado la originalidad de cada aportación multiplicándola por su elaboración. También se ha matizado la suma de las creatividades parciales multiplicándola por la flexibilidad mostrada en las aportaciones.

## 5 Conclusiones

Hemos presentado de forma detallada una evaluación de la creatividad de los alumnos en varias prácticas de una asignatura de algoritmos. Para cada práctica, se ha incluido su enunciado y un resumen de las contribuciones presentadas en cada informe, los resultados de los distintos componentes de creatividad y la creatividad global de cada grupo, se ha justificado el método de cálculo de cada componente y de la creatividad global, así como un debate. Los resultados han sido interesantes, tanto desde un punto de vista metodológico como por los resultados obtenidos.

**Agradecimientos.** Este trabajo se ha financiado con los proyectos de investigación TIN2015-66731-C2-1-R del Ministerio de Economía y Competitividad y S2018/TCS-4307 de la Comunidad Autónoma de Madrid. Este último también está financiado con fondos estructurales (FSE y FEDER).

## Referencias

1. Velázquez Iturbide, J.Á., Debdí, O., Esteban Sánchez, N., y Pizarro, C.: GreedEx: A visualization tool for experimentation and discovery learning of greedy algorithms. *IEEE Transactions on Learning Technologies*, 6, 2 (abril-junio 2013) 130-143, DOI [10.1109/TLT.2013.8](https://doi.org/10.1109/TLT.2013.8)
2. Esteban Sánchez, N., Pizarro, C., y Velázquez Iturbide, J.Á.: Evaluation of a didactic method for the active learning of greedy algorithms. *IEEE Transactions on Education*, 57, 2 (2014) 83-91, DOI [10.1109/TE.2013.2275154](https://doi.org/10.1109/TE.2013.2275154)
3. Velázquez Iturbide, J.Á., Hernán Losada, I., y Pérez Carrasco, A.: A «multiple executions» technique of visualization. En: Proceedings of the 21<sup>st</sup> Annual Conference on Innovation



- and Technology in Computer Science Education, ITiCSE 2016, ACM Press (2016) 59-64, DOI [10.1145/2899415.2899451](https://doi.org/10.1145/2899415.2899451)
4. Velázquez Iturbide, J.Á.: Exploring the joint use of educational theories and information technology to improve CS courses. En: IEEE Global Engineering Education Conference, EDUCON 2017, IEEE (2017), 1.561-1.570, DOI [10.1109/EDUCON.2017.7943057](https://doi.org/10.1109/EDUCON.2017.7943057)
  5. Velázquez Iturbide, J.Á.: An experimental method for the active learning of greedy algorithms. ACM Transactions on Computing Education, 13, 4 (octubre 2013) artículo 18, DOI [10.1145/2534972](https://doi.org/10.1145/2534972)
  6. Velázquez Iturbide, J.Á.: Identification and removal of misconceptions on optimization concepts underlying greedy algorithms. Journal of Research and Practice in Information Technology, 45, 3/4 (agosto 2013) 203-217
  7. Velázquez Iturbide, J.Á.: Difficulties, attitudes and misconceptions on experimenting with optimization algorithms. En: Proceedings of the 2014 International Symposium on Computers in Education (SIIE'14), IEEE Xplore (2014) 17-22, DOI [10.1109/SIIE.2014.7017698](https://doi.org/10.1109/SIIE.2014.7017698)
  8. Velázquez Iturbide, J.Á.: Students' misconceptions of optimization problems. En: Proceedings of the 24<sup>th</sup> Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2019, ACM Press (2019) 464-470, DOI [10.1145/3304221.3319749](https://doi.org/10.1145/3304221.3319749)
  9. Velázquez Iturbide, J.Á., Hernán Losada, I., y Paredes Velasco, M.: Evaluating the effect of program visualization on student motivation. IEEE Transactions on Education, 60, 3 (2017) 238-245
  10. Martí Oliet, N., Ortega Mallén, Y., y Verdejo, A.: Estructuras de datos y métodos algorítmicos. Ibergarceta Publicaciones, 2<sup>a</sup> ed., 2013
  11. Kleinberg, J., Tardos, É.: Algorithm Design, Pearson Addison-Wesley, 2006
  12. Brassard, G., Bratley, P.: Fundamentals of Algorithmics. Englewood Cliffs, NJ: Prentice-Hall, 1996
  13. Torrance, E.P.: The nature of creativity as manifest in its testing. En: The Nature of Creativity: Contemporary Psychological Perspectives, Sternberg, R. J. (ed.). Nueva York: University Press, 1988, 148-176

## Apéndice A: Enunciado de las Prácticas

### Grados en Ingeniería Informática e Ingeniería de Computadores Asignatura *Algoritmos Avanzados* Curso 2019/2020 Práctica nº 2

#### Objetivo

El objetivo de la práctica es que el alumno practique con los algoritmos heurísticos.

#### Carácter

La realización de la práctica es voluntaria, pero su entrega es un requisito para poder entregar la práctica 3b. Puede hacerse individualmente o en pareja.

#### Enunciado

Un equipo de informáticos va contratando y realizando trabajos semanalmente. Para organizarse mejor, dividen los trabajos entre los que conllevan poca tensión y los que suponen gran tensión. Si realizan un trabajo de baja tensión en la semana  $i$ -ésima, cobran una cantidad  $b_i > 0$  euros mientras que por un trabajo de alta tensión cobran  $a_i > 0$  euros. Un trabajo de baja tensión puede contratarse en cualquier semana pero, para poder aguantar la tensión, un trabajo de alta tensión solamente se contrata para la semana  $i$  si han descansado en la semana  $i-1$  (con la única excepción de la semana 0, en la cual puede escogerse un trabajo de alta tensión). Por tanto, en una semana cualquiera puede realizarse un trabajo de baja tensión, un trabajo de alta tensión o descansar.

El *problema del plan de trabajos estresantes de beneficio máximo* consiste en decidir, dada una secuencia de beneficios  $b_i$  y  $a_i$ , qué secuencia de trabajos debe contratarse en  $n$  semanas para obtener un beneficio máximo. Aunque los trabajos de alta tensión suelen dar un beneficio mayor que los de baja tensión, no tiene por qué ser necesariamente así.

Por ejemplo, sean los siguientes beneficios para 5 semanas:

	Semana 0	Semana 1	Semana 2	Semana 3	Semana 4
$a$	10	15	15	15	15
$b$	10	10	10	10	10

Un plan de trabajos con beneficio máximo sería tomar los cinco trabajos de baja tensión, con un beneficio igual a  $10+10+10+10+10=50$ . Puede comprobarse que no hay ningún otro plan que proporcione un beneficio mayor.

Un algoritmo heurístico H1 elige trabajos mediante la comparación sucesiva, desde  $i=0$  y cada dos semanas  $i$  e  $i+1$ , entre los valores  $b_i+b_{i+1}$  y  $a_{i+1}$ .

Sea el ejemplo anterior de nuevo. El algoritmo H1 elige los dos trabajos de baja tensión en las semanas 0 y 1 (con beneficio  $b_0+b_1=10+10=20$ , superior a  $a_1=15$ ), vuelve a hacer lo mismo en las semanas 2 y 3 (de nuevo, con beneficio  $10+10=20$ ) y finalmente elegirá el trabajo de baja tensión  $b_4=10$ . El beneficio total es 50, que es un valor maximal.

El objetivo de la práctica es implementar al menos dos algoritmos heurísticos y usar OptimEx para comparar su optimalidad, medida como el porcentaje de casos en los que cada algoritmo calcula una solución optimal con respecto al otro algoritmo.

## Informe

El alumno debe entregar un informe con la estructura que se detalla a continuación. El código de los algoritmos no debe enviarse en ficheros separados, sino integrarse en el texto del informe. Sólo se debe incluir el código de los métodos pedidos y, si es el caso, de los métodos auxiliares que aquéllos utilicen.

1. **Especificación del problema.** Debe desarrollarse una especificación del problema, identificando sus distintas partes: precondition y poscondition (formada a su vez por condición de validez y función objetivo). No hace falta utilizar notación matemática, pero la respuesta debe ser precisa y completa.
2. **Implementación de los algoritmos.** Implementar el algoritmo heurístico antes esbozado con la siguiente cabecera:

```
public static int trabajosH1 (int[] a, int[] b)
```

donde  $a[i]$  es el precio del trabajo de alta tensión de la semana  $i$ -ésima,  $0 \leq i \leq n-1$ , y  $b[i]$  es el precio del trabajo de baja tensión de la misma semana.

También se pide idear e implementar uno o más algoritmos heurísticos adicionales, que sean “razonables”. Asimismo, para cada nuevo algoritmo, se explicará claramente el criterio voraz usado y una justificación intuitiva de por qué dicho criterio puede conducir a soluciones optimales. La cabecera de estos nuevos algoritmos debe ser igual a la anterior (aunque, obviamente, con distintos identificadores de método).

3. **Experimentación con la optimalidad de los algoritmos.** Se comparará con OptimEx la optimalidad de los algoritmos desarrollados. Se aconseja leer detalladamente la guía de usuario (incompleta) disponible en el campus virtual. Debe aportarse la siguiente información:
  - Rangos de valores usados para la generación aleatoria de los datos de entrada.
  - Decidir si algún algoritmo es exacto.

- Evidencias en las que se basa la conclusión anterior: tabla de resumen numérico y diagramas del resumen gráfico.
  - Si algún algoritmo es inexacto, aportar dos contraejemplos, uno encontrado en el experimento y otro más sencillo deducido tras analizar las características de los contraejemplos encontrados experimentalmente. Explica las características del segundo contraejemplo que producen su suboptimalidad.
4. **Conclusiones.** Se presentan las conclusiones obtenidas tras realizar la práctica. Estas conclusiones pueden consistir en una valoración de los algoritmos heurísticos o cualquier comentario sobre la práctica. Por ejemplo, pueden describirse las incidencias que han dificultado la realización de la práctica, sus aspectos más atractivos o más difíciles, sugerencias sobre cómo mejorar la práctica, etc.

### **Entrega**

El alumno debe entregar el informe por medio del apartado de Evaluación del campus virtual. Si se tienen dificultades, puede enviarse por el correo del campus virtual con el asunto "Práctica 2". El plazo de entrega es el domingo 6 de octubre de 2019, incluido.

### **Evaluación**

Se evaluará la calidad y claridad de todos los apartados del informe.

**Grados en Ingeniería Informática e Ingeniería de  
Computadores  
Asignatura *Algoritmos Avanzados*  
Curso 2019/2020  
Práctica nº 3a**

**Objetivo**

El objetivo de la práctica es que el alumno profundice en su conocimiento de las técnicas de búsqueda en espacios de estados.

**Carácter**

La realización de la práctica es voluntaria. Puede hacerse individualmente o en pareja.

**Enunciado**

En la práctica 2 se planteó el *problema del plan de trabajos estresantes de beneficio máximo*. Se dispone de ofertas de trabajos para  $n$  semanas, unos de alta tensión y beneficio  $b_i$  y otros de baja tensión y beneficio  $a_i$ ,  $0 \leq i \leq n-1$ . Recuérdese que el enunciado del problema imponía restricciones para la contratación de un trabajo de alta tensión. El objetivo consiste en contratar una secuencia de trabajos a contratar para las  $n$  semanas de forma que el beneficio obtenido sea máximo.

Por ejemplo, dados los siguientes beneficios para 5 semanas:

	Semana 0	Semana 1	Semana 2	Semana 3	Semana 4
$a$	10	15	15	15	15
$b$	10	10	10	10	10

un plan de trabajos con beneficio máximo consiste en contratar los cinco trabajos de baja tensión, con un beneficio igual a  $10+10+10+10+10=50$ .

El objetivo de la práctica es desarrollar *de forma sistemática* un algoritmo de vuelta atrás que resuelva el problema planteado y comprobar experimentalmente su optimalidad.

**Informe**

El alumno debe entregar un informe con la estructura que se detalla a continuación. El código de los algoritmos no debe enviarse en ficheros separados, sino integrarse en el texto del informe. Sólo se debe incluir el código de los métodos pedidos y, si es el caso, de los métodos auxiliares que aquéllos utilicen.

1. **Técnica de vuelta atrás.** Constará de varios apartados, que mostrarán el desarrollo sistemático de un algoritmo de vuelta atrás:

- a) Diseño de un árbol de búsqueda adecuado para resolver el problema. Debe incluirse una figura del mismo, acompañada de una explicación breve y clara, en función de los parámetros del problema, de: (1) el número de niveles del árbol, y (2) los candidatos existentes en cada nodo del árbol. Si el árbol es muy grande, basta con mostrar una parte, siempre que sea fácil deducir la parte omitida.
- b) Comprobación de validez. Debe explicarse breve y claramente la comprobación de validez de la solución parcial realizada en cada nodo del árbol de búsqueda.
- c) Código de un algoritmo de vuelta atrás, que incorpore el diseño anterior. Al igual que en la práctica 2, la cabecera del método principal del algoritmo debe ser:

```
public static int trabajosVA (int[] a, int[] b)
```

donde  $a[i]$  es el precio del trabajo de alta tensión de la semana  $i$ -ésima,  $0 \leq i \leq n-1$ , y  $b[i]$  es el precio del trabajo de baja tensión de la misma semana.

Para el ejemplo anterior, la llamada del método principal será `trabajosVA({10,15,15,15,15},{10,10,10,10,10})`. El algoritmo debe devolver el beneficio de un plan óptimo de trabajos; opcionalmente, puede imprimir dicho plan óptimo.

2. **Comparación de optimalidad.** Se añadirá el algoritmo de vuelta atrás a la clase utilizada en la práctica 2 (con dos algoritmos heurísticos, al menos) y se comparará la optimalidad de todos los algoritmos. Obsérvese que el algoritmo de vuelta atrás debe ser exacto, es decir, debería calcular resultados óptimos en el 100% de los casos. Si no fuera así, deberían revisarse todos los algoritmos para identificar y corregir los errores, y repetir el experimento hasta que se obtengan los resultados predichos por la teoría. (En el manual de OptimEx puede consultarse un catálogo de situaciones conflictivas y su resolución.) Se pide:
  - a) Material del experimento. Se debe:
    - Identificar cada algoritmo, indicando la técnica de diseño con la que se ha desarrollado y, si es necesario, algún elemento diferenciador (p.ej. la función de selección de cada algoritmo heurístico).
    - Indicar los rangos de valores usados para la generación aleatoria de los datos de entrada.
  - b) Conclusión. Se dirá qué algoritmos son exactos según los resultados de la experimentación.
  - c) Evidencias. Deben aportarse los resultados recogidos en la tabla de resumen numérico y en los diagramas de resumen gráfico, explicando su significado.
  - d) Incidencias (opcional). Si durante la realización de esta práctica ha sido necesario revisar y modificar algún algoritmo de prácticas anteriores, se explicará por qué se realizó dicha modificación y en qué consistió.
3. **Conclusiones.** Se explican las conclusiones obtenidas tras realizar la práctica. Estas conclusiones pueden consistir en una valoración de las técnicas de búsqueda o cualquier otro comentario sobre la práctica (incidencias que han dificultado la realización de la práctica, sus aspectos más atractivos o más difíciles, sugerencias sobre cómo mejorar la práctica, etc.)

**Entrega**

El alumno debe entregar el informe por medio del apartado de Evaluación del campus virtual. Si se tienen dificultades, puede enviarse por el correo del campus virtual con el asunto "Práctica 3a". El plazo de entrega del informe es el domingo 20 de octubre de 2019, incluido.

**Evaluación**

Se evaluará la calidad y claridad de todos los apartados del informe.

**Grados en Ingeniería Informática e Ingeniería de  
Computadores  
Asignatura *Algoritmos Avanzados*  
Curso 2019/2020  
Práctica nº 3b**

**Objetivo**

El objetivo de la práctica es que el alumno profundice en su conocimiento de las técnicas de búsqueda en espacios de estados.

**Carácter**

La realización de la práctica es voluntaria. Debe hacerse en el mismo grupo que la práctica 3a (es decir, individualmente o con el mismo compañero).

**Enunciado**

En la parte (a) de la práctica 3 se pidió desarrollar de forma sistemática un algoritmo de vuelta atrás que resolviera el *problema del plan de trabajos estresantes de beneficio máximo* y comprobar su optimalidad. El objetivo de esta segunda parte es similar, pero referido a un algoritmo de ramificación y poda. Obsérvese que ambos algoritmos calcularán el mismo resultado, aunque previsiblemente de forma más eficiente en el caso de ramificación y poda.

**Informe**

El alumno debe entregar un informe con la estructura que se detalla a continuación. El código de los algoritmos no debe enviarse en ficheros separados, sino integrarse en el texto del informe. Sólo se debe incluir el código de los métodos pedidos y, si es el caso, de los métodos auxiliares que aquéllos utilicen.

1. **Técnica de vuelta atrás (opcional).** Este apartado sólo se rellenará si se realiza alguna modificación en el algoritmo presentado para la práctica 3a. En este caso, deben explicarse breve y claramente los cambios realizados usando los apartados previstos:
  - a) Árbol de búsqueda.
  - b) Condición de validez.
  - c) Código del algoritmo de vuelta atrás.
2. **Técnica de ramificación y poda.** Partiendo del algoritmo de vuelta atrás anterior, se pide:



- a) Función de cota para este problema. Debe proponerse al menos una función de cota. Para cada una, debe proporcionarse: (1) definición de la función de cota en términos de los parámetros del problema, (2) valor inicial de la cota, y (3) cómo se actualiza su valor en cada nodo del árbol de búsqueda.
  - b) Código de un algoritmo basado en la técnica de ramificación y poda, que incorpore una de las cotas anteriores. El algoritmo debe tener la misma cabecera para el método principal que el algoritmo de vuelta atrás.
3. **Comparación de optimalidad.** Se añadirá el algoritmo de ramificación y poda a los algoritmos de la clase utilizada en la práctica 3a y se comparará la optimalidad de todos los algoritmos. Obsérvese que todos los algoritmos de búsqueda deben ser exactos, es decir, deberían calcular resultados óptimos en el 100% de los casos. Si no fuera así, deberían revisarse todos los algoritmos para identificar y corregir los errores, y repetir el experimento hasta que se obtengan los resultados predichos por la teoría. (En el manual de OptimEx puede consultarse un catálogo de situaciones conflictivas y su resolución.) Se pide:
- a) Material del experimento. Se debe:
    - Identificar cada algoritmo, indicando la técnica de diseño con la que se ha desarrollado y, si es necesario, algún elemento diferenciador (p.ej. la función de selección de cada algoritmo heurístico).
    - Indicar los rangos de valores usados para la generación aleatoria de los datos de entrada.
  - b) Conclusión. Se dirá qué algoritmos son exactos según los resultados de la experimentación.
  - c) Evidencias. Deben aportarse los resultados recogidos en la tabla de resumen numérico y en los diagramas de resumen gráfico, explicando su significado.
  - d) Incidencias (opcional). Si durante la realización de esta práctica ha sido necesario revisar y modificar algún algoritmo de prácticas anteriores, se explicará por qué se realizó dicha modificación y en qué consistió.
4. **Comparación de eficiencia en tiempo.** Se repetirá el experimento con los mismos algoritmos y datos que en el apartado anterior, pero seleccionando en OptimEx el criterio de eficiencia en tiempo (en lugar de optimalidad), es decir, se compararán los tiempos de ejecución de los algoritmos. Se pide:
- a) Conclusión. Se comentarán breve y claramente las diferencias en tiempo de ejecución de los algoritmos, según los resultados de la experimentación.
  - b) Evidencias. Deben aportarse los tiempos medidos en la tabla de resumen numérico y en los diagramas de resumen gráfico, explicando su significado.
5. **Conclusiones.** Se explican las conclusiones obtenidas tras realizar la práctica. Estas conclusiones pueden consistir en una valoración de las técnicas de búsqueda o cualquier otro comentario sobre la práctica (incidencias que han dificultado la realización de la práctica, sus aspectos más atractivos o más difíciles, sugerencias sobre cómo mejorar la práctica, etc.)

## Entrega

El alumno debe entregar el informe por medio del apartado de Evaluación del campus virtual. Si se tienen dificultades, puede enviarse por el correo del campus virtual con el asunto "Práctica 3b". El plazo de entrega del informe es el lunes 28 de octubre de 2019, incluido.

### **Evaluación**

Se evaluará la calidad y claridad de todos los apartados del informe.

**Grados en Ingeniería Informática e Ingeniería de  
Computadores  
Asignatura Algoritmos Avanzados  
Curso 2019/2020  
Práctica nº 6**

**Objetivo**

El objetivo de la práctica es que el alumno practique con los algoritmos probabilistas.

**Carácter**

La realización de la práctica es voluntaria. Puede hacerse individualmente o en pareja.

**Enunciado**

Sea la versión de minimización del *problema del llenado de cajas*. Se dispone de varios objetos, cada uno con un peso  $p_i$ , y un suministro ilimitado de cajas de capacidad  $c$ . El objetivo consiste en introducir los objetos en un número mínimo de cajas.

Por ejemplo, sean cinco objetos de pesos  $\{5,2,4,1,8\}$  y una capacidad  $c=10$  de las cajas. Pueden introducirse los objetos en 2 cajas si en la primera se meten los objetos de pesos  $\{5,4,1\}$  y en la segunda, los objetos de pesos  $\{2,8\}$ .

Pueden encontrarse en el Aula Virtual varios algoritmos que resuelven este problema, tanto en el paquete de algoritmos aproximados como en el paquete de algoritmos de búsqueda. La cabecera de estos métodos es:

```
public static int menosCajas (int[] ps, int c)
```

Dado que es un problema de minimización, una forma de abordar el problema consiste en comenzar sin ninguna caja y que se vaya usando una caja nueva cada vez que sea necesario. Cada objeto nuevo podría introducirse en una caja ya disponible que tenga suficiente espacio libre o en una caja nueva. El problema puede resolverse de forma probabilista si la decisión de dónde introducir cada objeto se toma al azar entre una caja nueva o cualquiera de las cajas ya disponibles y con capacidad libre suficiente.

El objetivo de la práctica es implementar al menos un algoritmo probabilista que resuelva el problema y usar OptimEx para comparar su optimalidad.

**Informe**

El alumno debe entregar un informe con la estructura que se detalla a continuación. El código de los algoritmos no debe enviarse en ficheros separados, sino integrarse en el texto del informe. Sólo se debe incluir el código de los métodos pedidos y, si es el caso, de los métodos auxiliares que aquéllos utilicen.

1. **Desarrollo del algoritmo.** Explicar e implementar el algoritmo probabilista antes esbozado, respetando la cabecera anterior.
2. **Experimentación con la optimalidad del algoritmo.** Se comparará con OptimEx la optimalidad del algoritmo probabilista y algunos algoritmos disponibles en el Aula Virtual para este problema (al menos, los algoritmos heurísticos y el algoritmo de vuelta atrás). Debe aportarse la siguiente información:
  - a) Material del experimento. Se debe:
    - Identificar cada algoritmo, indicando la técnica de diseño con la que se ha desarrollado y, si es necesario, algún elemento diferenciador (p.ej. la función de selección de cada algoritmo heurístico).
    - Indicar los rangos de valores usados para la generación aleatoria de los datos de entrada.
  - b) Conclusión. Se dirá qué algoritmos son exactos según los resultados de la experimentación.
  - c) Evidencias. Deben aportarse los resultados recogidos en la tabla de resumen numérico y en los diagramas de resumen gráfico, explicando su significado.
  - d) Incidencias (opcional). Si durante la realización de esta práctica ha sido necesario revisar y modificar algún algoritmo, se explicará por qué se realizó dicha modificación y en qué consistió.
3. **Mejora (opcional).** Pueden aportarse ideas de cómo mejorar la optimalidad del algoritmo probabilista anterior o de algoritmos probabilistas basados en un diseño distinto, e incluso programarlos y medir su optimalidad.
4. **Conclusiones.** Se presentan las conclusiones obtenidas tras realizar la práctica. Estas conclusiones pueden consistir en una valoración de los algoritmos heurísticos o cualquier comentario sobre la práctica. Por ejemplo, pueden describirse las incidencias que han dificultado la realización de la práctica, sus aspectos más atractivos o más difíciles, sugerencias sobre cómo mejorar la práctica, etc.

### Entrega

El alumno debe entregar el informe por medio del apartado de Evaluación del campus virtual. Si se tienen dificultades, puede enviarse por el correo del campus virtual con el asunto “Práctica 6”. El plazo de entrega es el domingo 15 de diciembre de 2019, incluido. No habrá segunda entrega.

### Evaluación

Se evaluará la calidad y claridad de todos los apartados del informe.