

J. Ángel Velázquez Iturbide
Isidoro Hernán Losada

Evaluación de la Eliminación de la Recursividad con SRec

Número 2015-05

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	Protocolo de la Evaluación	1
3	Análisis y Resultados	3
3.1	Método de Análisis.....	3
3.2	Resultados.....	3
4	Resumen de Hallazgos.....	9
5	Conclusiones.....	10
	Agradecimientos.....	10
	Referencias.....	11
	Apéndice: Enunciado de la Práctica de la Sexta Evaluación	12

Evaluación de la Eliminación de la Recursividad con SRec

J. Ángel Velázquez Iturbide, Isidoro Hernán Losada

Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos,
C/ Tulipán s/n, 28933, Móstoles, Madrid
{angel.velazquez,isidoro.hernan}@urjc.es

Resumen. SRec es un sistema para la visualización de la recursividad. Este informe presenta los resultados de una evaluación del uso docente de SRec para eliminar la recursividad redundante, realizada como parte de una práctica. Se describe el protocolo de evaluación utilizado y los resultados obtenidos. En los resultados se observan pautas comunes entre el uso o no de SRec, pero también algunas diferencias claras. El informe incluye, como apéndice, el enunciado de la práctica.

Palabras clave: Recursividad redundante, eliminación redundancia, SRec, árbol de recursión, grafo de dependencia, tabla.

1 Introducción

SRec es un sistema de visualización de la recursividad [1]. Es un sistema altamente interactivo concebido como apoyo a la docencia de los algoritmos [2]. Se han desarrollado varias extensiones, orientadas a técnicas de diseño específicas (divide y vencerás [3] y programación dinámica [4][5]).

A lo largo del ciclo de vida de SRec se han realizado diversas evaluaciones de usabilidad [6][7]. En este informe se presenta una evaluación del uso docente de sus funciones para análisis y eliminación de la recursividad redundante, realizada como parte de una práctica. La estructura del informe es la siguiente. El apartado 2 describe el protocolo utilizado. El apartado 3 presenta el método de análisis y los resultados obtenidos. En el cuarto apartado resumimos los hallazgos de la evaluación. Finalmente, un apéndice recoge el enunciado de la práctica.

2 Protocolo de la Evaluación

Esta evaluación de SRec se realizó en noviembre de 2015, en la asignatura optativa “Algoritmos Avanzados”, de cuarto curso del Grado en Ingeniería Informática. Participaron alumnos del grupo presencial del campus de Móstoles.

Los alumnos habían recibido dos sesiones de clase dedicadas al análisis y eliminación de la redundancia en algoritmos recursivos y una sesión de introducción a la programación dinámica.

La evaluación se realizó aprovechando la sesión de laboratorio de la práctica 4, cuyo objetivo es el análisis y eliminación de la recursividad redundante. La sesión tenía una duración de dos horas. La realización de la práctica era individual. Durante la sesión también se realizó una evaluación de motivación. No se esperaba que los alumnos pudieran acabar la práctica durante la sesión, por lo que tenían un plazo de una semana.

Veamos la organización de la sesión. Primero se les explicó que la participación en las evaluaciones era voluntaria y que el objetivo era mejorar la docencia de la asignatura. Ningún alumno rehusó participar. Después, contestaron a un pretest de motivación, accedieron al material disponible en el campus virtual, realizaron lo que les dio tiempo de la práctica y contestaron a un postest de motivación. Al final de la sesión, debían entregar el resultado de su trabajo bien subiendo un fichero al campus virtual bien en papel. Aunque no es el informe final de la práctica, por concreción llamaremos informe a esta entrega.

Los alumnos se dividieron en dos grupos (grupo experimental y grupo de control), diferenciados por que el grupo experimental usó SRec mientras que el grupo de control podía usar cualquier otro medio. Cada grupo estaba en un aula informática distinta.

Veamos las diferencias de tratamiento entre ambos grupos:

1. Grupo experimental. Al comienzo de la sesión, el profesor hizo una presentación rápida de SRec, que incluía: configuración inicial, carga y procesamiento de clases, selección de método, lanzamiento de ejecución, animación, control de visualización de árbol de recursión, búsqueda de nodos redundantes, grafo de dependencia, tabla y exportación. Los alumnos del grupo tenían disponible SRec junto al enunciado de la práctica en el campus virtual. Al final de la sesión, debían rellenar un cuestionario de usabilidad [8].
2. Grupo de control. Al final de la sesión, el profesor hizo la misma presentación rápida de SRec que en el grupo experimental, para que tuvieran los mismos medios para completar la práctica.

Asistieron 17 alumnos del grupo experimental y 18 del grupo de control. Cuatro alumnos del grupo experimental que usaban los ordenadores del laboratorio tuvieron problemas en usar SRec, mientras que los que usaban sus propios ordenadores portátiles no tuvieron problemas destacados. De estos cuatro alumnos, tres entregaron el resultado. Consideramos las entregas de estos 3 alumnos como pertenecientes al grupo de control. En total, hubo 2 alumnos del grupo experimental que no entregaron nada. Por otro lado, un alumno del grupo de control utilizó una versión anterior de SRec (de la que disponía tras su paso por la asignatura "Diseño y Análisis de Algoritmos"). Consideramos la entrega de este alumno como perteneciente al grupo experimental.

Incluimos el enunciado de la práctica en el Apéndice.

3 Análisis y Resultados

Primero presentamos el método de análisis para presentar a continuación los resultados del análisis. Se recogieron 13 cuestionarios del grupo experimental y 20 del grupo de control.

3.1 Método de Análisis

Una primera diferencia entre ambos grupos, previsible por el desarrollo de la sesión, era el material de entrega. Se informó a los alumnos del grupo de control de que podrían entregarlo en papel, dado que no disponían de las facilidades de SRec para la generación y exportación de las visualizaciones. La Tabla 1 contiene la forma de entrega en ambos grupos.

Tabla 1. Medio de entrega de los informes

	Grupo experimental	Grupo de control
Papel	–	8 (40%)
Fichero	13 (100%)	8 (40%)
Ambos	–	4 (20%)
Total de entregas	13	20

Las entregas mixtas constaban de un fichero que incluía bien una fotografía realizada con el móvil de cada dibujo realizado bien dejaba un hueco para cada dibujo entregado en papel.

Para el análisis se siguió el siguiente proceso:

1. Se hizo una primera identificación de aspectos clave a partir de los elementos que los alumnos aportaban en sus informes (distintas visualizaciones, algoritmos, etc.).
2. Primera ronda. Durante la misma se vio la necesidad de reubicar a algunos alumnos de ambos grupos, dadas las incidencias antes señaladas.
3. Segunda ronda. Dada la experiencia de la primera ronda, se detallaron más diversos aspectos, sobre todo relacionados con los árboles de recursión.
4. Tercera ronda. Durante la elaboración del informe se vio la conveniencia de analizar con más detalle la parte de los grafos de dependencia y tablas.

3.2 Resultados

Recordemos que se recogieron 13 cuestionarios del grupo experimental y 20 del grupo de control. Sin embargo, una de las entregas del grupo de control consistió en un fichero Java con el algoritmo recursivo propuesto y los algoritmos memorizado y tabulado pedidos; por tanto, no puede usarse para el análisis de la mayor parte de los aspectos.

Al analizar las representaciones gráficas producidas, parecía que una primera diferencia podría encontrarse en el tamaño de los árboles de recursión generados. La Tabla 2 muestra los parámetros usados en las distintas entregas.

Tabla 2. Parámetros usados para generar el árbol de recursión

	Grupo experimental		Grupo de control	
	#	%	#	%
2, 1	–	–	2	10,53%
2, 2	7	53,85%	8	42,11%
2, 3	5	38,46%	7	36,84%
3, 3	1	7,69%	1	5,26%
4, 3	–	–	1	5,26%
Total	13	100%	19	100%

Puede observarse que el grupo de control produjo mayor variación en los parámetros, tanto el caso más pequeño como el mayor. Sin embargo, si asimilamos los dos casos más pequeños y los dos más grandes, no hay grandes diferencias porcentuales entre ambos grupos.

Otra diferencia esperable son las características gráficas del árbol. La Tabla 3 muestra los resultados obtenidos.

Tabla 3. Características del árbol de recursión generado

	Grupo experimental		Grupo de control	
	#	%	#	%
Completo	12	92,31%	13	68,42%
Incompleto	1	7,69%	5	26,32%
Mal	–	–	1	5,26%
Total	13	100%	19	100%

El árbol incompleto recogido en el grupo experimental era una parte recortada del árbol generado por SRec, pero suficiente para mostrar redundancia entre dos nodos. También es llamativo un árbol completo pero dibujado de forma totalmente libre (números y flechas), casi sin relación con el formato convencional.

Un árbol recogido en el grupo de control no correspondía a la ejecución sistemática del algoritmo. En cuanto a los 5 árboles incompletos: 2 omitían los descendientes de nodos redundantes, 1 incluía todos los nodos pero sin contenido en los descendientes de nodos redundantes, 1 usaba puntos suspensivos para los descendientes de nodos redundantes y 1 usaba puntos suspensivos para algunos subárboles no desarrollados.

Otro paso importante en la práctica era el análisis de redundancia. La Tabla 4 presenta muestra la forma de presentar la redundancia en las figuras.

Tabla 4. Formas de representar la redundancia

	Grupo experimental		Grupo de control	
	#	%	#	%
Nodos	7	53,85%	10	52,63%
Subárboles	5	38,46%	3	15,79%
Explicada	–	–	1	5,26%
Ninguna	1	7,69%	5	26,32%
Total	13	100%	19	100%

De los 7 alumnos del grupo experimental que presentaron la redundancia basada en nodos repetidos, 5 usaron las funciones de SRec de búsqueda de nodos o resaltado de nodos, mientras que otros 2 lo hicieron manualmente sobre la visualización. Parece que 4 de estos alumnos usaron la función de búsqueda, mientras que otro fue resaltando nodo a nodo (ya que no aparecen resaltados todos los nodos de ningún valor concreto). Asimismo, los 5 alumnos que presentaron la redundancia basada en subárboles repetidos realizaron un dibujo (p.ej. cuadrados o círculos) sobre la visualización generada por SRec.

Resumiendo el uso de SRec para esta tarea por parte del grupo experimental, hubo 5 alumnos (41'67%) que usaron SRec y 7 alumnos (58'33%) que no lo usaron. También podemos afirmar que 4 alumnos usaron la función de búsqueda (33'33%) mientras que 8 (66'67%) no lo hicieron.

Los dos alumnos del grupo de control que habían generado el árbol correspondiente a los parámetros (2,1), marcaron casos básicos repetidos, dado que este caso tan sencillo no contiene llamadas recursivas repetidas. Un alumno identificó de forma escrita casos de redundancia, pero sin reflejarlo en el árbol de recursión.

También se examinó si la redundancia se comprobó sobre un único valor de los parámetros o se comprobó sobre varios.

Tabla 5. Número de valores distintos de los parámetros usados para detectar redundancia

	Grupo experimental		Grupo de control	
	#	%	#	%
1	9	75%	8	57,14%
Más de 1	3	25%	6	42,86%
Total	12	100%	14	100%

Puede comprobarse que el porcentaje de alumnos que comprobaron la redundancia con más de un nodo es mayor en el grupo de control que en el experimental.

Por último, se examinó el grado de exhaustividad mostrado por los alumnos que no habían usado la función de búsqueda de nodos de SRec, es decir, se examinó si buscaron todos los nodos que repetían unos valores o se confirmaban con encontrar dos nodos.

Tabla 6. Exhaustividad en la identificación de todos los nodos redundantes sin usar SRec

	Grupo experimental		Grupo de control	
	#	%	#	%
Exhaustivo	4	50%	12	85,71%
No exhaustivo	4	50%	2	14,29%
Total	7	100%	14	100%

Puede observarse que el grupo de control fue más exhaustivo en esta comprobación.

La segunda representación gráfica tratada en esta práctica fueron los grafos de dependencia. La Tabla 7 muestra las apariciones de grafos de dependencia en los informes entregados por los alumnos, diferenciando entre los generados con SRec o los dibujados manualmente.

Tabla 7. Generación de los grafos de dependencia

	Grupo experimental		Grupo de control	
	#	%	#	%
Con SRec	12	92,31%	–	–
Sin SRec	1	7,69%	18	90%
No	–	–	2	10%
Total	13	100%	20	100%

Puede comprobarse que todos los alumnos del grupo experimental generaron un grafo de dependencia, aunque uno de ellos sin usar SRec. En el grupo de control, casi todos los producen (90%, manualmente).

Dado que los alumnos del grupo de control han generado el grafo de dependencia sin SRec, hay más incidencias que destacar. Dos alumnos incluyen flechas entre casos básicos. Otros 2 alumnos han borrado flechas entre casos básicos. Un alumno tiene una flecha que enlaza nodos equivocados. En resumen, un porcentaje destacado de las entregas ha cometido errores (5 de 19, 27'8%), aunque algunos los han arreglado. Por último, un alumno ha presentado un grafo de dependencia para valores distintos que en su árbol de recursión (más sencillo, con datos 2,1, ya que el árbol de recursión era sobre un caso más complejo 4, 3, que no desarrolló completamente).

La tercera representación gráfica implicada en la práctica fue la tabla necesaria para la eliminación de la redundancia. La Tabla 8 muestra las apariciones de tablas en los informes entregados por los alumnos, diferenciando entre las generadas con SRec o las dibujadas manualmente.

Tabla 8. Generación de las tablas

	Grupo experimental		Grupo de control	
	#	%	#	%
Con SRec	8	61,54%	–	–
Sin SRec	4	30,77%	17	85,00%
No	1	7,69%	3	15,00%
Total	13	100%	20	100%

Puede comprobarse que casi todos los alumnos de ambos grupos presentan una tabla. En el grupo experimental, el porcentaje de aquellos que la generan sin usar SRec es más destacado que en representaciones anteriores (30'77%).

De los 8 alumnos que usaron SRec, solamente 1 utilizó la función de tabular los nodos (especificando expresiones). Los restantes 7 alumnos usaron la función (mucho más sencilla) de dibujar la tabla sobre el grafo de dependencia (especificando el número de filas y de columnas). Dos de ellos primero recolocaron los nodos y uno de éstos generó una tabla posterior para otros datos (y quizá con la función de tabular).

También puede resultar interesante conocer la representación de las tablas realizadas por los alumnos que no han usado SRec. Un detalle es si en cada celda de la tabla se han representado los parámetros de las llamadas recursivas, sus resultados o ambos. La Tabla 9 resume lo realizado por los alumnos.

Tabla 9. Contenido de las tablas

	Grupo experimental		Grupo de control	
	#	%	#	%
Llamadas	–	–	6	35,29%
Resultados	4	100%	2	11,76%
Ambos	–	–	8	47,06%
Patrón de dependencia	–	–	1	5,88%
Total	4	100%	14	100%

Resulta curioso que, mientras todos los alumnos del grupo experimental que no usaron SRec representaron únicamente los resultados de las llamadas recursivas, sólo interesó esta información a un 58'8% de los alumnos del grupo de control, frente a un 82'3% a los que les interesó más las llamadas realizadas.

Otro detalle es si han mantenido las dependencias sobre la representación tabular. La Tabla 10 indica qué alumnos (de los que no usaron SRec) mantuvieron las dependencias en el dibujo de la tabla.

Tabla 10. Conservación de dependencias en la tabla

	Grupo experimental		Grupo de control	
	#	%	#	%
Sí	3	75%	11	64,7%
No	1	25%	6	35,3%
Total	4	100%	14	100%

En ambos grupos es mayoritaria la proporción de alumnos que prefieren mostrar las dependencias que los que no.

Un último elemento pedido era la declaración de la tabla en Java. La Tabla 11 muestra lo realizado por los alumnos sobre este aspecto.

Tabla 11. Declaración de la tabla en Java

	Grupo experimental		Grupo de control	
	#	%	#	%
Tabla en Java	11	91,67%	11	55,00%
Tabla general	–	–	3	15,00%
Tabla para un ejemplo	–	–	5	25,00%
Sin declaración	1	8,33%	1	5,00%
Total	12	100%	20	100%

Recogemos 20 respuestas del grupo de control porque los 3 alumnos que no aportaron dibujo de la tabla, codificaron un algoritmo memorizado, del cual podemos conocer el tamaño (en Java) de la misma. Los alumnos de la categoría “general” han identificado el número de filas y columnas de la tabla en la representación gráfica de la misma, pero no lo han codificado en Java. En todo caso, es una tarea trivial, por lo que ambas categorías suman 70%.

Aunque no se les pedía explícitamente, podemos comprobar si han realizado otros productos de la práctica. Realizaron algún elemento extra 5 alumnos del grupo experimental (de 13, 38'5%) y 7 del grupo de control (de 20, 35%). La Tabla 12 muestra las entregas realizadas.

Tabla 12. Otras fases realizadas de la práctica

	Grupo experimental	Grupo de control
Algoritmo memorizado completo	4	5
Algoritmo memorizado, método principal	1	1
Algoritmo memorizado, árbol de recursión	2	–
Algoritmo memorizado, análisis de complejidad	2	–
Algoritmo tabulado	2	2
Algoritmo tabulado, análisis de minimización de memoria	–	1
Algoritmo tabulado, análisis de complejidad	2	–
Total	13	9

Obsérvese que los alumnos del grupo experimental que han entregado elementos extras, han entregado más elementos que sus compañeros del grupo de control.

Este aparentemente mayor rendimiento del grupo experimental tiene correspondencia en la observación realizada por los profesores de ambos grupos, que percibieron que los alumnos del grupo experimental iban abandonando el aula informática mientras que apenas abandonaban los alumnos del grupo de control.

Si comprobamos la hora de entrega en el campus virtual, obtenemos los resultados mostrados en la Tabla 13. (Hay que advertir que los tiempos de entrega del grupo experimental deberían tender a ser posteriores porque sus alumnos recibieron una sesión inicial de explicación de SRec, con lo que comenzaron a realizar la práctica después que el grupo de control, y a los alumnos del grupo de control se les urgió al final a entregar la práctica porque debían recibir su sesión posterior de explicación de SRec.)

Tabla 13. Entrega de los informes en el campus virtual

	Grupo experimental	Grupo de control
Número de entregas	13	12
Hora de la primera entrega	16:16	16:27
Hora de la última entrega	17:00	16:54
Hora media de entrega	16:37	16:45

4 Resumen de Hallazgos

Podemos resumir los resultados de la evaluación como sigue:

- Es algo mayor el porcentaje de alumnos del grupo experimental que han presentado una representación gráfica que en el grupo de control: árbol de recursión (100% vs. 95%), grafo de dependencia (100% vs. 90%) y tabla (92'3% vs. 85%).
- Los árboles presentados en ambos grupos son de tamaño similar, pequeño. Los alumnos del grupo de control presentan árboles con mayor variedad de tamaños, a veces simplificados.
- Junto a la identificación de nodos redundantes, es frecuente encontrar la identificación de subárboles redundantes en ambos grupos. Sin embargo, el grupo de control presenta un mayor porcentaje de alumnos que no han analizado la redundancia (26'3% vs. 7'7%) y, dentro de los que la han analizado, de los que han sido exhaustivos en la identificación de redundancia (85'7% vs. 42'9%).
- Algunos alumnos del grupo de control presentan errores en sus grafos de dependencias (27'8%).
- En el grupo experimental hay un porcentaje de alumnos mayor que no han usado SRec para generar las tablas que las representaciones anteriores (30'8% vs. 0% y 7'7%). Los alumnos que usan SRec utilizan mayoritariamente (87'5%) la función de dibujar la tabla en lugar de tabular el grafo de dependencia; esto significa que han practicado poco con la función de tabular.
- Los alumnos que han dibujado la tabla manualmente prefieren mantener las dependencias entre celdas. El grupo experimental prefiere representar los resultados de las llamadas (100%), mientras que el grupo de control prefiere representar las llamadas (82'3%) que sus resultados (58'8%).
- Los alumnos del grupo experimental han sido más precisos, declarando la tabla en Java (91'7%), mientras que en el grupo de control (55%) es frecuente dar su tamaño sin declararla en Java o simplemente dibujándola para el grafo de dependencia (40%).
- Los alumnos del grupo experimental han sido más productivos, tanto por la cantidad de productos entregados como por el tiempo medio de entrega electrónica. En el primer caso, han entregado algoritmos el mismo porcentaje de alumnos, pero los del grupo experimental han entregado más algoritmos. Ne

el segundo caso, los alumnos del grupo experimental han entregado sus informes unos 7 minutos antes, de media (a pesar de meter más prisa a los del grupo experimental para explicarles la nueva versión de SRec).

Dando una interpretación cualitativa a estos hallazgos, podríamos afirmar lo siguiente:

- El comportamiento de ambos grupos es similar en numerosos aspectos:
 - Uso de las distintas representaciones gráficas
 - Uso de datos de entrada de pequeño tamaño.
 - Frecuente asociación de la redundancia con un subárbol, más que con un nodo.
- El grupo experimental (alumnos que usaron SRec) presenta algunas características:
 - Uso más frecuente del análisis de redundancia.
 - El uso de SRec favorece la continuación del trabajo en fases posteriores (declaración de la tabla en Java, algoritmos no redundantes, etc.).
 - El uso de SRec favorece la realización del trabajo en menos tiempo.
- El grupo de control (alumnos que no usaron SRec) presenta algunas características:
 - Frecuente simplificación de los árboles de recursión mediante varios recursos (puntos suspensivos, omitir subárboles o datos).
 - Cuando se analiza la redundancia, hacen un análisis más exhaustivo.
 - Mayor porcentaje de errores en la conversión de árbol a grafo.
- Sobre el uso de SRec:
 - Conviene revisar la función de búsqueda de nodos redundantes, al menos permitiendo que el usuario pueda elegir entre resaltar el nodo encontrado o todo su subárbol.
 - El uso eficiente de SRec en todas las etapas exige entrenamiento en todas las funciones, sobre todo la de tabulación.

5 Conclusiones

Hemos presentado de forma detallada una evaluación de SRec para la eliminación de la recursividad redundante realizada en noviembre de 2015. Se ha incluido el procedimiento usado, el método de análisis, los resultados y una breve valoración de los mismos. Los resultados han sido positivos para conocer las fortalezas y debilidades del uso y no uso de SRec.

Agradecimientos. Este trabajo se ha financiado con los proyectos TIN2011-29542-C02-01 del Ministerio de Economía y Competitividad de España y S2013/ICE-2715 de la Comunidad Autónoma de Madrid.

Referencias

1. Velázquez-Iturbide, J.Á., Pérez-Carrasco, A., Urquiza-Fuentes, J.: SRec: An animation system of recursion for algorithm courses. En: Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2008. ACM Press, New York (2008) 225–229, DOI 10.1145/1384271.1384332
2. Velázquez-Iturbide, J.Á., Pérez-Carrasco, A.: InfoVis interaction techniques in animation of recursive programs. *Algorithms* 3, 1 (2010) 76-91, DOI 10.3390/a3010076
3. Velázquez-Iturbide, J.Á., Pérez-Carrasco, A., Urquiza-Fuentes, J.: A design of automatic visualizations for divide-and-conquer algorithms. *Electronic Notes in Theoretical Computer Science* 224 (2009) 159-167, DOI 10.1016/j.entcs.2008.12.060
4. Velázquez-Iturbide, J.Á., Pérez-Carrasco, A.: Familias de visualizaciones de los árboles de recursión. En: SIIE13 XV International Symposium on Computers in Education – Proceedings, M.J. Marcelino, M.C. Azebedo Gomes y A.J. Mendes (eds.) (2013) 18-23
5. Velázquez-Iturbide, J.Á., Pastor Herranz, D., Pérez-Carrasco, A.: La visualización interactiva como apoyo al desarrollo de algoritmos de programación dinámica. En: Atas do XVII Simpósio Internacional de Informática Educativa (SIIE'15), M.R. Rodrigues, M. Llamas Nistal y M. Figueiredo (eds.) (2015) 195-201
6. Velázquez-Iturbide, J.Á., Pérez-Carrasco, A., Urquiza-Fuentes, J.: Multiple usability evaluations of a program animation tool. En: The 10th IEEE International Conference on Advanced Learning Technologies, M. Jemni, Kinshuk, D. Sampson y J.M. Spector (eds.) (2010) IEEE Computer Society, 452-454, DOI 10.1109/ICALT.2010.131
7. Velázquez Iturbide, J.Á., Pérez Carrasco, A., Debdi, O.: Experiences in usability evaluation of educational programming tools. En: Student Usability in Educational Software and Games: Improving Experiences, C. González (ed.), IGI Global (2013) 241-260, DOI 10.4018/978-1-4666-1987-6
8. Velázquez Iturbide, J.Á.: Evaluaciones sexta y séptima de usabilidad de SRec. En: Serie de Informes Técnicos DLSII-URJC, nº 2015-04, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos, 2015, 41 págs.

Apéndice: Enunciado de la Práctica de la Sexta Evaluación

Grado en Ingeniería Informática Asignatura *Algoritmos Avanzados* Curso 2015/2016 Práctica nº 4

Objetivo

El objetivo de la práctica es que el alumno practique la eliminación de la recursividad múltiple redundante.

Carácter

La práctica es voluntaria. Debe realizarse individualmente.

Enunciado

Sea el siguiente método:

```
public static int f (int x, int y) {
    if (x==0)
        return y;
    else if (y==0)
        return x;
    else
        return f(x,y-1) + f(x-1,y) + f(x-1,y-1);
}
```

Se pretende eliminar la redundancia existente en este algoritmo recursivo múltiple, obteniéndose dos algoritmos optimizados: uno mediante memorización y otro mediante tabulación.

Entrega

Debe entregarse un informe completo elaborado siguiendo el índice detallado a continuación. Este informe debe enviarse 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 4”. El plazo de entrega del informe es el jueves 19 de noviembre de 2015 a las 15h.

Esta práctica contiene una participación voluntaria en una actividad docente, que el profesor explicará. Tal y como aparece en las normas de la asignatura, los alumnos que acepten participar tendrán un incremento de 0’25 puntos sobre la nota de la asignatura ya aprobada. Dichos alumnos deberán seguir las instrucciones del profesor sobre actividades a realizar y entregas de materiales (por medio del apartado de Evaluación del campus virtual):

- Informe parcial con los puntos 1 y 2, al menos, del índice dado a continuación.
- Cuestionarios.

Informe

El alumno debe entregar un informe con la siguiente estructura. Para los apartados donde se pide una figura, si el sistema SRec (disponible en el campus virtual) soporta su generación, debe incluirse un gráfico generado con SRec; si no, puede producirse de cualquier otra forma (manualmente o con otro programa).

1. **Análisis de la redundancia.** Se incluirán representaciones gráficas representativas del comportamiento de la función f :
 - a. Árbol de recursión. Para comprobar que el algoritmo es redundante, se resaltará algún nodo correspondiente a un caso recursivo y que aparezca varias veces repetido.
 - b. Grafo de dependencia correspondiente.
2. **Diseño de la tabla.** Se pide diseñar una tabla adecuada para poder hacer posteriormente la memorización o tabulación del algoritmo recursivo:
 - a. Una figura con la tabla donde se haya colocado cada nodo del grafo de dependencia anterior en una celda distinta, manteniendo los arcos del grafo.
 - b. Sabiendo que el problema debe resolverse con un método que tenga la cabecera del enunciado, declara en Java una tabla t de tamaño adecuado (expresada en función de los parámetros) para la memorización o tabulación del algoritmo.
3. **Memorización.** Se desarrollará un algoritmo optimizado con la técnica de memorización. Debe incluirse:
 - a. Código.
 - b. Árbol de recursión generado por este algoritmo para los mismos datos de entrada usados en el apartado 1(a). También deben resaltarse los nodos que correspondan al mismo caso que en el apartado 1(a). (Si se considera conveniente para que la figura sea legible, pueden omitirse algunos parámetros.)
 - c. Análisis detallado de su complejidad en tiempo y en espacio.
4. **Tabulación.** Los mismos apartados (a) y (c) del punto anterior, pero aplicando la técnica de tabulación.
5. **Conclusiones.** Se explican las conclusiones obtenidas tras realizar la práctica. Estas conclusiones pueden consistir en una comparación de la eficiencia de los algoritmos, la experiencia de uso de SRec 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.)

Evaluación

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