

J. Ángel Velázquez Iturbide

**Dos Evaluaciones de la
Visualización con SRec de
Algoritmos de Búsqueda**

Número 2024-04

**Serie de Informes Técnicos DLSI1-URJC
ISSN 1988-8074
Departamento de Informática y Estadística
Universidad Rey Juan Carlos**

Índice

1	Introducción	1
2	Protocolo de la Primera Evaluación	2
3	Resultados de la Primera Evaluación	3
3.1	Visualización de la Técnica de Vuelta Atrás.....	3
3.2	Visualización de la Técnica de Ramificación y Poda.....	4
3.3	Vista de Líneas.....	5
3.4	Resumen de Hallazgos	5
4	Protocolo de la Segunda Evaluación	6
5	Resultados de la Segunda Evaluación	6
5.1	Visualización de la Técnica de Vuelta Atrás.....	6
5.2	Visualización de la Técnica de Ramificación y Poda.....	7
5.3	Vista de Líneas.....	8
5.4	Resumen de Hallazgos	8
6	Conclusiones	9
	Agradecimientos	10
	Referencias.....	10
	Apéndice A: Enunciado de la Primera Sesión de Familiarización	11
	Apéndice B: Cuestionario de Opinión de la Primera Evaluación.....	24
	Apéndice C: Respuestas al Cuestionario de la Primera Evaluación.....	30
	Apéndice D: Enunciado de la Segunda Sesión de Familiarización.....	33
	Apéndice E: Cuestionario de Opinión de la Segunda Evaluación.....	42
	Apéndice F: Respuestas al Cuestionario de la Segunda Evaluación	48

Dos Evaluaciones de la Visualización con SRec de Algoritmos de Búsqueda

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. SRec es un sistema para la visualización de la recursividad. Este informe presenta los resultados de dos evaluaciones de las visualizaciones que SRec genera para algoritmos de búsqueda en espacios de estados en forma de árbol, estudiando su aplicación a dos técnicas de diseño (vuelta atrás y ramificación y poda). Las evaluaciones se realizaron en una sesión de laboratorio de familiarización con el sistema SRec. En ambas evaluaciones se valoraron dos visualizaciones: una genérica (árboles de recursión) y otra diseñada específicamente para esta clase de algoritmos (vistas de líneas). En la primera evaluación también se indagó qué esquemas de programación de ambas técnicas de diseño eran más legibles y generaban visualizaciones más comprensibles. El informe incluye, como apéndices, el enunciado de cada sesión de familiarización, los cuestionarios utilizados y las respuestas de los alumnos.

Palabras clave: Recursividad, visualización de programas, SRec, vuelta atrás, ramificación y poda, árboles de recursión, vistas de líneas, cuestionarios.

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], programación dinámica [4] y últimamente las técnicas de diseño de búsqueda en espacios de estados en forma de árbol [5], entre ellas vuelta atrás y ramificación y poda.

En este informe se presentan dos evaluaciones de las visualizaciones para algoritmos de búsqueda. La primera evaluación era más exploratoria porque, a diferencia de la visualización de otras técnicas de diseño, había incertidumbre sobre la forma de codificar los algoritmos que facilitara simultáneamente su aprendizaje y la generación de visualizaciones. En la segunda evaluación solamente se valora las visualizaciones generadas. En ambas, solamente se evalúa la vista global de líneas, no la de rama.

La estructura del informe es la siguiente. Los apartados 2 y 3 describen, respectivamente, el protocolo utilizado y los resultados obtenidos en la primera

evaluación. Análogamente, los apartados 4 y 5 describen, respectivamente, el protocolo utilizado y los resultados obtenidos en la segunda evaluación. En el apartado 6 realizamos una recapitulación de los resultados obtenidos en ambas evaluaciones. Finalmente, seis apéndices recogen información detallada de las sesiones de evaluación, los cuestionarios y las respuestas de los alumnos en ambas evaluaciones.

2 Protocolo de la Primera Evaluación

Esta evaluación de SRec se realizó en noviembre de 2022, en la asignatura optativa “Algoritmos Avanzados”, de cuarto curso del Grado en Ingeniería Informática. La evaluación se realizó en una sesión de laboratorio, posterior a la práctica 3 (sobre las dos técnicas de búsqueda), que servía para la familiarización con SRec, que se usaría en las prácticas 5 y 6.

Al comienzo de la sesión, el profesor explicó que los materiales de la sesión estaban disponibles en el aula virtual. La sesión constaba de cinco partes:

- Las partes 1-2 servían de familiarización propiamente dicha con SRec.
- La parte 3 tenían el objetivo de comparar la claridad de dos codificaciones alternativas, y sus visualizaciones correspondientes, para algoritmos de vuelta atrás. Una codificación es más eficiente [6], mientras que la otra es más intuitiva, al seguir la estructura usual de los algoritmos recursivos. Ambas codificaciones se acompañaban con la visualización generada por SRec para unos datos concretos, estando identificados en el documento los parámetros mostrados en cada nodo.
- La parte 4 tenían el objetivo de comparar la claridad de dos codificaciones alternativas, y sus visualizaciones correspondientes, para algoritmos de ramificación y poda. Una codificación es más eficiente, realizando las podas tras generar cada candidato, mientras que la otra las retrasaba hasta realizar la siguiente llamada recursiva, permitiendo así mostrar en el árbol de recursión los casos podados. Ambas codificaciones se acompañaban con la visualización generada por SRec para los mismos datos que en la parte anterior, sin identificar en el documento los parámetros mostrados en cada nodo.
- La parte 5 repetía la segunda visualización generada para el algoritmo de ramificación y poda, numerando sus nodos, e incluía la vista global de líneas correspondiente.
- La parte 6 era de realización libre y podía ser tenida en cuenta para la nota final de la asignatura.

Incluimos el enunciado de la sesión en el Apéndice A, el cuestionario de opinión en el Apéndice B y los resultados del mismo en el Apéndice C.

3 Resultados de la Primera Evaluación

Presentamos los resultados del cuestionario separados en tres partes: esquema de codificación de vuelta atrás, esquema de codificación de ramificación y poda, y vista de líneas. Pueden encontrarse todas las respuestas en el Apéndice C.

3.1 Visualización de la Técnica de Vuelta Atrás

Se recogieron 11 cuestionarios. Veamos los resultados de las preguntas de esta primera parte:

- Los 11 alumnos probaron a ejecutar SRec con los datos de entrada proporcionados.
- A pesar de que el enunciado identificaba explícitamente los tres parámetros que se mostraban en cada nodo del árbol de recursión, solamente 4 alumnos (P05, P06, P07, P10) identificaron los sucesivos valores del parámetro $bOpt$, mientras que los otros 7 tomaron los valores devueltos por el método. Sin embargo, los alumnos se fijaron en la primera visualización, obteniendo la secuencia de valores mejores -1,1,17,23, cuando la secuencia completa es -1,0,1,16,17,23,26.
- Solamente 5 alumnos identifican claramente cuándo se produce la poda de nodos inválidos, frente a 6 alumnos que lo hacen erróneamente. De estos, 4 alumnos responden que es 1 solo nodo, mientras que otros dos alumnos responden 2 y 7.
- Solamente 2 alumnos prueban los algoritmos con datos propios. En un caso, usan una variación de los datos del ejemplo, mientras que otro alumno usa datos sin relación con el enunciado.
- La Tabla 1 muestra la valoración de las visualizaciones y las codificaciones:
 - Solamente 2 alumnos dan un valor distinto entre sí a las dos visualizaciones, aunque en sentido contrario. Otros 7 alumnos las valoran igual, mientras que 3 solamente valoran la primera (en este caso, también suponemos que dan el mismo valor a ambas).
 - Las dos visualizaciones obtienen el mismo valor medio, pero la segunda presenta más variación. Ambas visualizaciones reciben un valor alto, cercano a 4.
 - Solamente 1 alumno da un valor distinto a las dos codificaciones, resultando algo mejor la primera codificación. Ambas visualizaciones reciben un valor alto, cercano a 4.

Tabla 1. Resultados para la visualización de los algoritmos de vuelta atrás, primera evaluación

Pregunta	Media	Desv. típica	Mediana	Moda	Mínimo	Máximo
Visualización 1	3,91	0,94	4	3	3	5
Visualización 2	3,91	1,04	4	5	2	5
Codificación 1	3,91	0,70	4	4	3	5
Codificación 2	3,82	0,75	4	4	3	5

- No hay comentarios libres.

3.2 Visualización de la Técnica de Ramificación y Poda

Se recogieron 9 cuestionarios, ya que dos participantes en la parte primera (P02, P03) no entregaron este cuestionario. Veamos los resultados de las preguntas de esta segunda parte:

- 8 de los 9 alumnos afirman haber ejecutado SRec con los datos de entrada proporcionados. Dado que el otro alumno responde a las preguntas siguientes, lo lógico es suponer que también lo hizo.
- El enunciado no mencionaba explícitamente los tres parámetros que se mostraban en cada nodo del árbol de búsqueda (*bParc*, *bOpt* y *cota*). Por los valores que aportan los alumnos, responden fijándose en el primer árbol. 4 alumnos (P06, P07, P09, P10) responden correctamente que se producen 2 podas (aunque P10 da, curiosamente, un valor menor en una unidad de las dos cotas). Otros 2 alumnos (P01, P04) responden los 6 nodos de hoja, mientras que otro responde 3 nodos de hoja (P08). Finalmente, un alumno (P11) responde los 5 nodos en los que se producen un corte (sean por invalidez, sea por cota), mientras que otro (P05) responde 3 (que son los valores distintos mostrados de la cota en todos los nodos donde se produce una poda).
- Solamente 1 alumno (P08) afirma haber probado los algoritmos con datos propios, pero no los aporta.
- La Tabla 2 muestra la valoración de las visualizaciones y las codificaciones. Incluimos solamente la valoración de una visualización y una codificación porque todos los alumnos dan el mismo valor a las dos visualizaciones y a las dos codificaciones.
 - 6 alumnos valoran igual las dos visualizaciones, mientras que 3 solamente valoran la primera visualización.
 - Los mismos resultados se recogen al valorar las dos codificaciones.
 - La visualización de ramificación y poda es valorada casi igual que la de vuelta atrás, pero su codificación se valora más.

Tabla 2. Resultados para la visualización de los algoritmos de ramificación y poda, primera evaluación

Pregunta	Media	Desv. típica	Mediana	Moda	Mínimo	Máximo
Visualización	3,89	0,78	4	4	3	5
Codificación	4,22	0,97	5	5	3	5

- No hay comentarios libres.

3.3 Vista de Líneas

Se recogieron 8 cuestionarios, ya que otro participante (P09) entregó el cuestionario en blanco. Veamos los resultados de las preguntas de esta tercera parte:

- 6 de los 8 alumnos afirman haber probado las vistas de líneas, ejecutando SRec con los datos de entrada proporcionados.
- Solamente 4 alumnos (P06, P07, P08, P10) ven la correspondencia entre los nodos de hoja y los nodos de la vista de líneas con un rombo grande, mientras que 1 no la ve (P11). Dos alumnos (P04, P05) que respondieron al resto de preguntas de esta parte no respondieron a esta.
- El número de alumnos que ven la correspondencia entre los nodos de hoja y los nodos de la vista de líneas con una cruz baja a 3 (P06, P07, P08), habiendo 2 que declaran no verla (P10, P11). Se produce la misma abstención de dos alumnos.
- Solamente 1 alumno (P08, el mismo que en la parte anterior) afirma haber probado los algoritmos con datos propios, pero no los aporta.
- La Tabla 3 muestra la valoración de la vista de líneas. 6 alumnos valoran igual vista con puntuaciones altas (dos 5, cuatro 4).

Tabla 3. Resultados para la evolución de valores en la vista de líneas, primera evaluación

Media	Desv. típica	Mediana	Moda	Mínimo	Máximo
4,33	0,52	4	4	4	5

- 4 alumnos prefieren el uso combinado de los árboles de recursión y la vista de líneas. Un alumno prefiere los árboles de recursión y otro prefiere la vista de líneas.
- Hay un comentario, aparentemente sobre la utilidad de la vista (P11): “Ahora la cabecera tiene sentido para mí”.

3.4 Resumen de Hallazgos

Podemos resaltar las siguientes conclusiones:

- Pocos alumnos han interpretado correctamente las visualizaciones generadas, tanto para identificar los sucesivos valores de las soluciones mejores como las podas realizadas.
- Los alumnos se fijan más en la primera visualización de árbol de recursión producida para cada técnica. Al mostrar estas visualizaciones menos información, las respuestas dadas son incompletas.
- Pocos alumnos han utilizado datos propios para seguir familiarizándose con las visualizaciones.
- Las visualizaciones basadas en árboles de recursión y las codificaciones asociadas han recibido valores altos (cerca pero debajo de 4), destacando la

codificación de la técnica de ramificación y poda que visualiza los nodos podados.

- No se termina de comprender la correspondencia entre los nodos del árbol de recursión y la vista de líneas. Sin embargo, recibe una valoración muy alta (por encima de 4), siendo preferido su uso combinado con los árboles de recursión.

También disponemos de algunas notas tomadas por el investigador sobre el transcurso de la sesión, que podían ser útiles en una segunda evaluación:

- Conviene que el profesor haga una demostración de SRec al inicio de la sesión.
- En dicha demostración, se debe explicar explícitamente varios aspectos de SRec, entre ellos la instalación (por las restricciones que impone) y el filtrado de métodos y parámetros (necesario para visualizar algoritmos de búsqueda).
- Para los algoritmos de búsqueda, debe explicarse el procesado necesario para generarlas y dónde aparecen por defecto.

4 Protocolo de la Segunda Evaluación

La segunda evaluación fue bastante parecida a la anterior, por lo que no repetimos el protocolo seguido. El enunciado de la sesión recogió dos cambios principales, que solamente se aportó el segundo algoritmo (y su correspondiente visualización) de los aportados en cada una de las técnicas. EL cuestionario se simplificó coherentemente. Al comienzo de la sesión, el profesor también realizó una pequeña demostración del uso de SRec.

Incluimos el enunciado de la sesión en el Apéndice D, el cuestionario de opinión en el Apéndice E y los resultados del mismo en el Apéndice F.

5 Resultados de la Segunda Evaluación

A la sesión asistieron 13 alumnos, aunque solamente 8 enviaron contribuciones. Aunque el cuestionario tenía carácter individual, dos parejas de alumnos lo presentaron de forma conjunta (P05, P06), quedando un total de 6 cuestionarios recibidos.

Presentamos los resultados del cuestionario separados en tres partes: esquema de codificación de vuelta atrás, esquema de codificación de ramificación y poda, y vista de líneas. Pueden encontrarse todas las respuestas en el Apéndice C.

5.1 Visualización de la Técnica de Vuelta Atrás

Veamos los resultados de las preguntas de esta primera parte:

- Los 6 participantes probaron a ejecutar SRec con los datos de entrada proporcionados.
- A pesar de que el enunciado identificaba explícitamente los tres parámetros que se mostraban en cada nodo del árbol de recursión y que se mostraba una sola visualización, solamente 1 participante (P05) identificó correctamente los sucesivos valores del parámetro *bOpt*. Otros 4 participantes omitían algunos valores: -1 (P01), 26 (P02, P03), -1 y 0 (P06). Finalmente, otro participante (P04) da una solución extraña, quizá debida a haberse fijado solamente en la rama de la derecha (-1,23,23,23,23).
- Solamente 1 participante (P05) identifica claramente en cuántos casos se produce la poda de nodos inválidos (sin identificarlos). Otros 3 participantes (P01, P02, P03) contestan 7 nodos, mientras que otros 2 participantes (P04, P06) responden 2.
- Solamente 1 participante (P05) prueba los algoritmos con datos propios, sin apenas relación con el ejemplo del enunciado.
- La Tabla 4 muestra la valoración de la visualización y la codificación. En ambos casos, los valores obtenidos son muy altos (por encima de 4), siendo algo mayor la visualización que el código.

Tabla 4. Resultados para la visualización de los algoritmos de vuelta atrás, segunda evaluación

Pregunta	Media	Desv. típica	Mediana	Moda	Mínimo	Máximo
Visualización	4,33	0,52	4	4	4	5
Codificación	4,17	0,75	4	4	3	5

- No hay comentarios libres.

5.2 Visualización de la Técnica de Ramificación y Poda

Veamos los resultados de las preguntas de esta segunda parte:

- Los 6 participantes probaron a ejecutar SRec con los datos de entrada proporcionados.
- De nuevo, el enunciado no mencionaba explícitamente los tres parámetros que se mostraban en cada nodo del árbol de búsqueda (*bParc*, *bOpt* y *cota*). En esta evaluación, los 6 participantes responden correctamente, aunque con matices. Un participante (P04) contesta que se producen 3 podas, pero solamente identifica explícitamente 2; podría ser una errata. Otros 3 participantes (P01, P02, P03) dan la misma respuesta, incluyendo una identificación de decisiones que no corresponde con ninguna de las que sufren un corte.
- Solamente 1 participante (P05) prueba los algoritmos con datos propios, los mismos que en la primera parte.
- La Tabla 5 muestra la valoración de la visualización y la codificación. Se mantiene la tendencia de la parte primera, pero con la visualización aún mejor valorada.

Tabla 5. Resultados para la visualización de los algoritmos de ramificación y poda, segunda evaluación

Pregunta	Media	Desv. típica	Mediana	Moda	Mínimo	Máximo
Visualización	4,50	0,55	4, 5	4	4	5
Codificación	4,17	0,41	4	4	4	5

- No hay comentarios libres.

5.3 Vista de Líneas

Veamos los resultados de las preguntas de esta tercera parte:

- 5 de los 6 participantes afirman haber ejecutado SRec con los datos de entrada proporcionados. Dado que el otro participante responde a las preguntas siguientes, lo lógico es suponer que también lo hizo.
- 5 de los 6 participantes (P01, P02, P03, P05, P06) ven la correspondencia entre los nodos de hoja y los nodos de la vista de líneas con un rombo grande.
- Todos los 6 participantes ven la correspondencia entre los nodos de hoja y los nodos de la vista de líneas con una cruz, reconociendo que es un corte.
- Solamente 1 participante (P05) prueba los algoritmos con datos propios, los mismos que en la primera parte.
- La Tabla 6 muestra la valoración de la vista de líneas. Los valores son muy altos, aún más que la del árbol de recursión.

Tabla 6. Resultados para la evolución de valores en la vista de líneas, segunda evaluación

Media	Desv. típica	Mediana	Moda	Mínimo	Máximo
4,67	0,52	5,00	5	4	5

- 5 participantes prefieren el uso combinado de los árboles de recursión y la vista de líneas. El alumno que prefiere los árboles de recursión (P04) es el mismo que no vio la correspondencia entre nodos de hoja y nodos de la vista de líneas con un rombo grande.
- No hay comentarios libres.

5.4 Resumen de Hallazgos

Podemos resaltar las siguientes conclusiones:

- Las respuestas de los tres primeros participantes son iguales (salvo en algunas valoraciones), por lo que probablemente han trabajado conjuntamente. Obviamente, esta situación se ha repetido con las parejas consideradas participantes P05 y P06.

- La mayor parte de los participantes han interpretado incorrectamente la evolución de los mejores resultados y de las podas de nodos inválidos. Sin embargo, parecen haber interpretado correctamente las podas por cotas. Es necesario explicar explícitamente qué datos se muestran en los nodos de los árboles de recursión, así como interpretar el proceso de modificación de los datos de entrada y salida.
- Pocos alumnos han utilizado datos propios para seguir familiarizándose con las visualizaciones.
- Las visualizaciones basadas en árboles de recursión y las codificaciones asociadas han recibido valores muy altos (pero encima de 4), destacando la valoración del árbol de recursión generado para ramificación y poda.
- La vista de líneas recibe la valoración más alta de todas las visualizaciones (4'67), siendo preferido su uso combinado con los árboles de recursión.

6 Conclusiones

Hemos presentado de forma detallada dos evaluaciones de visualizaciones generadas por SRec para algoritmos de búsqueda, así como su codificación, realizadas en cursos académicos consecutivos. Se ha incluido el procedimiento y enunciado usado, los resultados detallados y comentados. Podemos resumir los hallazgos en:

- Parecen ser adecuadas las dos codificaciones utilizadas en la segunda evaluación: esquema de vuelta atrás sin optimizar y esquema de ramificación y poda que permite visualizar los nodos podados por cota. El esquema de vuelta atrás optimizado no muestra información relevante y hace más difícil de comprender todo lo que sucede durante el proceso de búsqueda.
- Para algoritmo de gran complejidad, como los de búsqueda, es necesario explicar los datos mostrados en los nodos los árboles de recursión, así como interpretar el proceso de modificación de los datos de entrada y salida. Obviamente, los alumnos deben estar familiarizados con las funciones de interacción que permiten llegar a estas visualizaciones, sobre todo las funciones de filtrado.
- La vista de árbol de recursión goza de gran aceptación, siendo la preferida cuando otras vistas no aportan claridad.
- Debe explicarse explícitamente la correspondencia entre los nodos del árbol de recursión y la vista de líneas También deben usarse de forma combinada ambas vistas.
- La vista de líneas tiene gran aceptación cuando se comprende.

De forma general, los resultados obtenidos han mejorado de la primera evaluación a la segunda, tras haber mejorado algo los materiales escritos y dejado las opciones aparentemente mejores de codificación y visualización. Sin embargo, los resultados de la segunda evaluación no pueden considerarse totalmente satisfactorios, por lo que es necesario mejorar aún más los materiales y el protocolo. Otra conclusión para estas

visualizaciones complejas es que no deberían mostrarse en la primera sesión, de familiarización, con SRec.

Agradecimientos. Este trabajo se ha financiado una ayuda al grupo de investigación de alto rendimiento LITE de la Universidad Rey Juan Carlos (ref. M3286).

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., 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
3. Velázquez-Iturbide, J.Á., Pérez-Carrasco, A.: Systematic development of dynamic programming algorithms assisted by interactive visualization. En: Proceedings of the 21st Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE'16. ACM Press, New York (2016) 71-76, DOI 10.1145/2899415.2899450
4. Velázquez-Iturbide, J.Á., Ivanov-Andreev, R.: Recursion-based visualizations of search algorithms in state-spaces. En: SIIE 2022, 2022 International Symposium on Computers in Education (SIIE), Gomes, A., Mendes, A. J., Caeiro Rodríguez, M., Nolan, S. (eds.), IEEE Xplore (2022), 6 págs., DOI 10.1109/SIIE56031.2022.9982362

Apéndice A: Enunciado de la Primera Sesión de Familiarización

Grados en Ingeniería Informática e Ingeniería de Computadores Asignatura *Algoritmos Avanzados* Curso 2022/2023 Sesión de familiarización con SRec

El objetivo de la sesión es la familiarización con el uso del sistema de visualización de programas SRec. SRec es una aplicación para la visualización de métodos recursivos, disponible en el aula virtual y que se utilizará para prácticas posteriores.

Asimismo, los apartados 3-5 de la sesión buscan valorar la comprensibilidad de varias alternativas de visualización de los algoritmos de vuelta atrás y de ramificación y poda. Para ello, se pide responder a varios cuestionarios cortos, de forma voluntaria. Las respuestas recogidas en el cuestionario servirán para mejorar la asignatura en cursos futuros.

Material

Para una realización cómoda de esta sesión se dispone de dos clases de Java, *varios.java* y *mochila01.java*.

Guion

La sesión consiste en la realización individual de las siguientes actividades, todas ellas voluntarias. Las dos primeras sirven de familiarización con SRec, mientras que las siguientes abordan la visualización de algoritmos de búsqueda.

La primera vez que se ejecuta SRec, debe indicarse un directorio donde se encuentre un fichero *java.exe* o *javac.exe* de un JDK de Java.

1. **Iniciación.** Sean los siguientes métodos para el problema de las torres de Hanoi. En conjunto, calculan el total de movimientos de discos a realizar, imprimiéndolos en el terminal:

```
public static int hanoi (int n) {
    System.out.println
        ("Van a moverse " + n + " discos");
    int mov = h (n, 'A', 'B', 'C');
    System.out.println
        ("Se han realizado " + mov + " movimientos");
    return mov;
}

private static int h (int n, char a, char b, char c) {
    int mov = 0;
    if (n!=0) {
        int mov1 = h (n-1, a,c,b);
        System.out.println ("se mueve un disco del poste "
            + a + " al poste " + c);
        int mov2 = h (n-1, b,a,c);
        mov = mov1 + 1 + mov2;
    }
    return mov;
}
```

Se pide usar las principales funciones de SRec:

- Cargar el fichero Java *varios.java*.
- Seleccionar la función *hanoi* y lanzar una o varias ejecuciones con valores de *n* comprendidos entre 3 y 6.
- Experimentar con los controles de animación, observando su efecto sobre las vistas de la recursividad que ofrece SRec:
 - o Traza.
 - o Pila de control.
 - o Árbol de activación. Si el árbol es muy grande, sólo se muestra una parte de la vista, pero se facilita la navegación por el mismo mediante un visor situado en la parte inferior de la vista del árbol.
 - o Terminal.
- Interacción con las vistas:
 - o Mover los paneles e incluso ocultarlos.
 - o Borrar o cerrar el terminal.

2. Interacción. Sean los siguientes métodos que calculan la suma de los elementos de un vector mediante la técnica de divide y vencerás:

```
public static int sumaArray (int[] v) {
    return sumaDyV (v, 0, v.length-1);
}

private static int sumaDyV (int[] v,
                            int inf, int sup) {

    if (inf==sup)
        return v[inf];
    else {
        int med = (inf+sup)/2;
        int suma1 = sumaDyV (v, inf, med);
        int suma2 = sumaDyV (v, med+1, sup);
        return suma1 + suma2;
    }
}
```

Se pide generar una animación cualquiera y usar las principales funciones de SRec de filtrado y configuración:

- Control de la cantidad de información mostrada en el árbol (métodos, entrada/salida, parámetros).
- Control del zoom.
- Control del formato gráfico de las 3 visualizaciones (traza, pila de control, árbol de activación).
- Cambiar las vistas de un panel a otro.
- Almacenar una o varias vistas en ficheros gráficos (pueden contemplarse con el visor de imágenes de Windows).

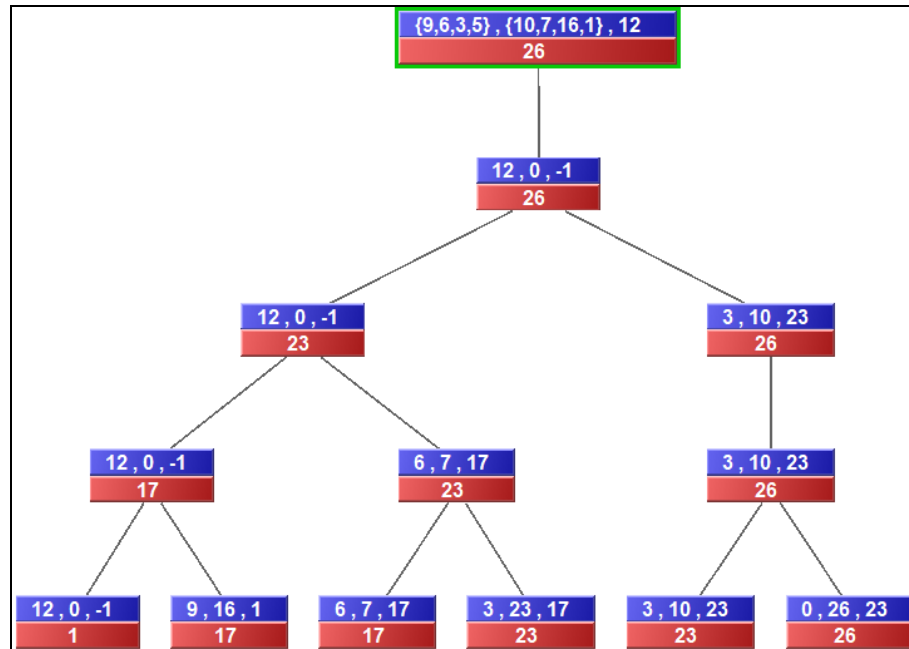
3. **Visualización de algoritmos de vuelta atrás.** Sea el conocido *problema de la mochila 0/1*. Se han desarrollado varias implementaciones de un algoritmo de vuelta atrás y de otro de ramificación y poda, que producen árboles de recursión ligeramente distintos. El objetivo de esta parte de la sesión es valorar la legibilidad de las visualizaciones producidas para su eventual uso en la asignatura en los cursos próximos.

Sea el algoritmo disponible en el fichero *Mochila_0_1.java*, conforme al esquema de codificación sugerido para la técnica algorítmica de vuelta atrás. Eliminando comentarios e instrucciones de entrada/salida, queda:

```
public static int mochila01_back_a
    (int[]ps, int[]bs, int c) {
    int[] solParcial = new int[ps.length];
    int[] solOptima = new int[ps.length];
    int bOpt =
        buscar01a(ps,bs,c,0,c,solParcial,0,solOptima,-1);
    return bOpt;
}

private static int buscar01a(int[] ps, int[] bs, int c,
    int i, int p,
    int[] solParc, int b,
    int[] solOpt, int bOpt) {
    for (int k=0; k<=1; k++) {
        if (k*ps[i]<=p) {
            solParc[i] = k;
            int np = p - k*ps[i];
            int nb = b + k*bs[i];
            if (i==ps.length-1) {
                if (nb>bOpt) {
                    bOpt = nb;
                    for (int j=0; j<ps.length; j++)
                        solOpt[j] = solParc[j];
                }
            }
            else
                bOpt = buscar01a
                    (ps,bs,c,i+1,np,solParc,nb,solOpt,bOpt);
        }
    }
    return bOpt;
}
```

En el resto del documento, usaremos los datos $ps=\{9,6,3,5\}$, $bs=\{10,7,16,1\}$, $c=12$, para los que el beneficio optimal es 26. El árbol de recursión que produce el algoritmo anterior es el siguiente, donde el nodo superior corresponde a la llamada al método principal *mochila01_back_a* y los demás nodos corresponden al método *buscar01a*, mostrando solamente los parámetros p , b , $bOpt$:



Sea otra versión, con el formato usual de los algoritmos recursivos:

```

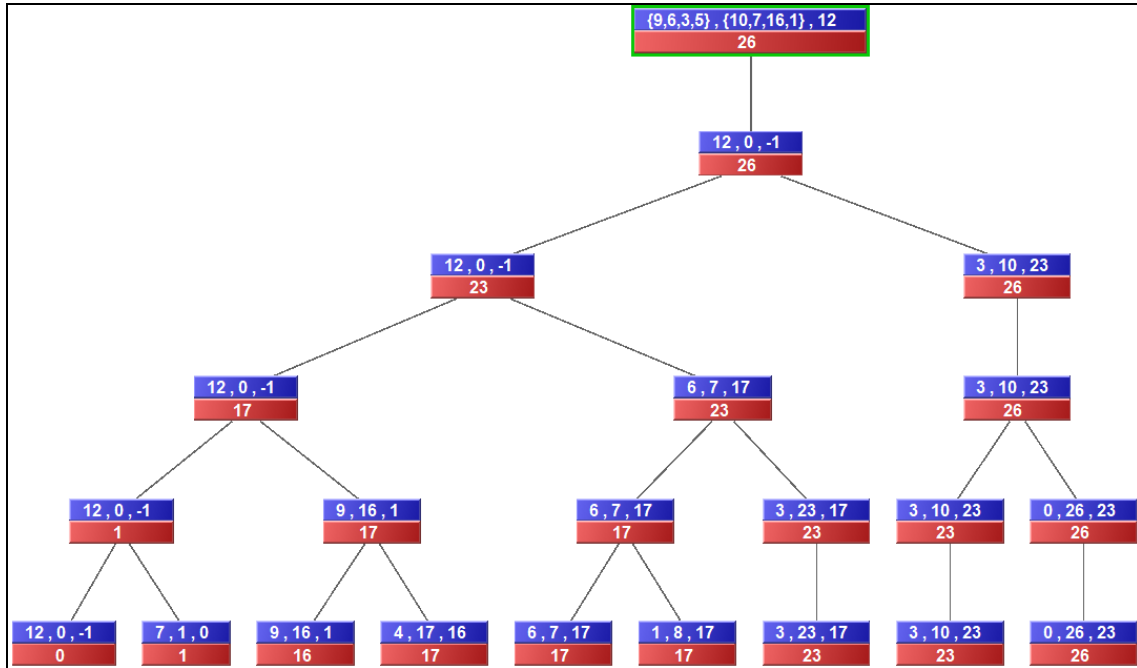
public static int mochila01_back_b
    (int[]ps, int[]bs, int c) {
    int[] solParcial = new int[ps.length];
    int[] solOptima = new int[ps.length];
    int bOpt =
        buscar01b(ps,bs,c,0,c,solParcial,0,solOptima,-1);
    return bOpt;
}

private static int buscar01b(int[] ps, int[] bs, int c,
    int i, int p,
    int[] solParc, int b,
    int[] solOpt, int bOpt) {

    if (i==ps.length) {
        if (b>bOpt) {
            bOpt = b;
            for (int j=0; j<ps.length; j++)
                solOpt[j] = solParc[j];
        }
    } else
        for (int k=0; k<=1; k++) {
            if (k*ps[i]<=p) {
                solParc[i] = k;
                int np = p - k*ps[i];
                int nb = b + k*bs[i];
                bOpt = buscar01b
                    (ps,bs,c,i+1,np,solParc,nb,solOpt,bOpt);
            }
        }
    return bOpt;
}

```

que genera el siguiente árbol de recursión, igual al anterior pero con un nivel más:



Por favor, contesta a las preguntas del apartado 3 del cuestionario. Si quieres, puedes probar primero SRec con estos mismos datos o con datos tuyos para familiarizarte mejor con ambos algoritmos y con las visualizaciones que generan.

4. **Visualización de algoritmos de ramificación y poda.** Sea ahora el siguiente algoritmo de ramificación y poda, cuya función de cota es igual a la suma de los beneficios. La comprobación de cota se realiza antes de las llamadas recursivas.

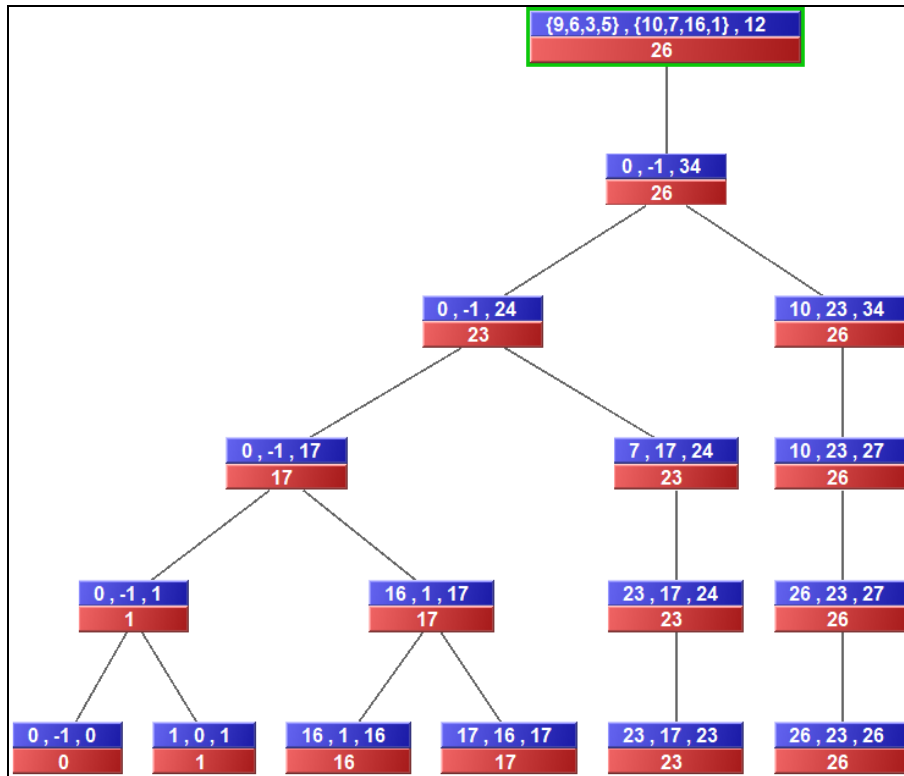
```

public static int mochila01_RyP_a
    (int[] ps, int[] bs, int c) {
    int[] solParcial = new int[ps.length];
    int[] solOptima = new int[ps.length];
    int cota = 0;
    for (int i=0; i<bs.length; i++)
        cota += bs[i];
    int bOpt = buscar01c
        (ps,bs,c,0,c,solParcial,0,solOptima,-1,cota);
    return bOpt;
}

private static int buscar01c(int[] ps, int[] bs, int c,
    int i, int p,
    int[] solParc, int b,
    int[] solOpt, int bOpt,
    int cota) {
    if (i==ps.length) {
        if (b>bOpt) {
            bOpt = b;
            for (int j=0; j<ps.length; j++)
                solOpt[j] = solParc[j];
        }
    } else
        for (int k=0; k<=1; k++) {
            if (k*ps[i]<=p) {
                solParc[i] = k;
                int np = p - k*ps[i];
                int nb = b + k*bs[i];
                int nCota = cota - (1-k)*bs[i];
                if (nCota>bOpt)
                    bOpt = buscar01c
                        (ps,bs,c,i+1,np,solParc,nb,solOpt,bOpt,nCota);
            }
        }
    return bOpt;
}

```

El árbol de recursión generado para el ejemplo anterior es:



Sea otra versión, en la que la comprobación de cota se realiza después de la llamada recursiva (como primera acción del método).

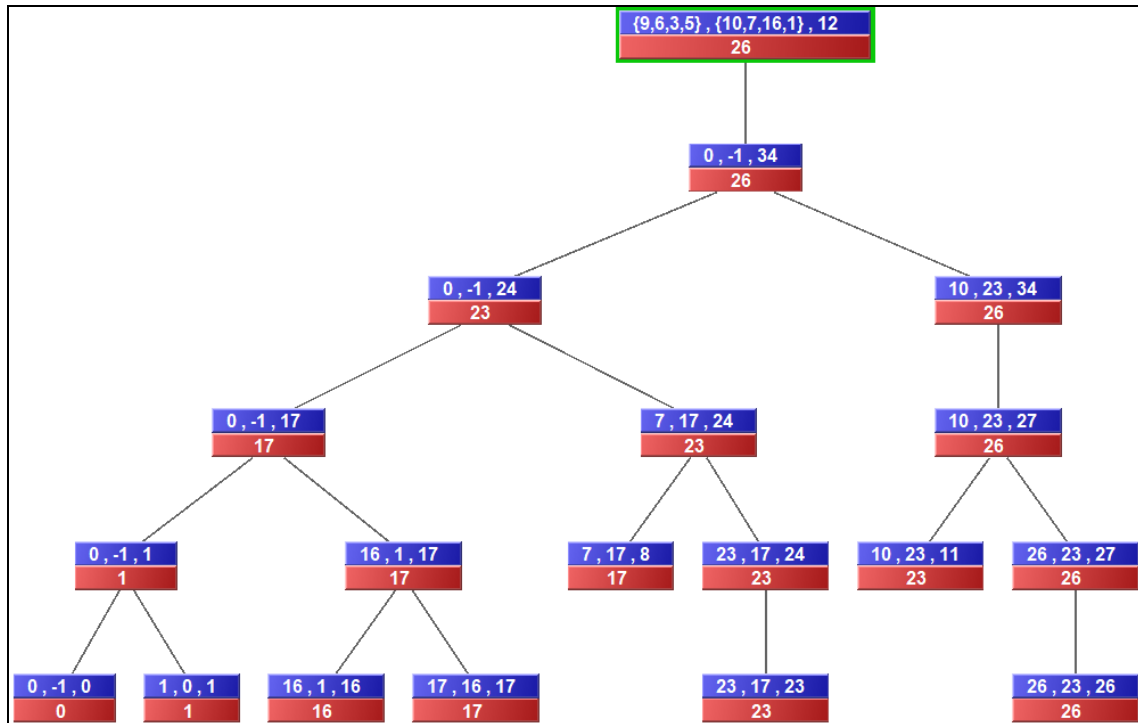
```

public static int mochila01_RyP_b
    (int[] ps, int[] bs, int c) {
    int[] solParcial = new int[ps.length];
    int[] solOptima = new int[ps.length];
    // cota como suma de beneficios
    int cota = 0;
    for (int i=0; i<bs.length; i++)
        cota += bs[i];
    int bOpt = buscar01d
        (ps,bs,c,0,c,solParcial,0,solOptima,-1,cota);
    return bOpt;
}

private static int buscar01d(int[] ps, int[] bs, int c,
    int i, int p,
    int[] solParc, int b,
    int[] solOpt, int bOpt,
    int cota) {
    if (cota>bOpt)
        if (i==ps.length) {
            if (b>bOpt) {
                bOpt = b;
                for (int j=0; j<ps.length; j++)
                    solOpt[j] = solParc[j];
            }
        } else
            for (int k=0; k<=1; k++) {
                if (k*ps[i]<=p) {
                    solParc[i] = k;
                    int np = p - k*ps[i];
                    int nb = b + k*bs[i];
                    int nCota = cota - (1-k)*bs[i];
                    bOpt = buscar01d
                        (ps,bs,c,i+1,np,solParc,nb,solOpt,bOpt,nCota);
                }
            }
    return bOpt;
}

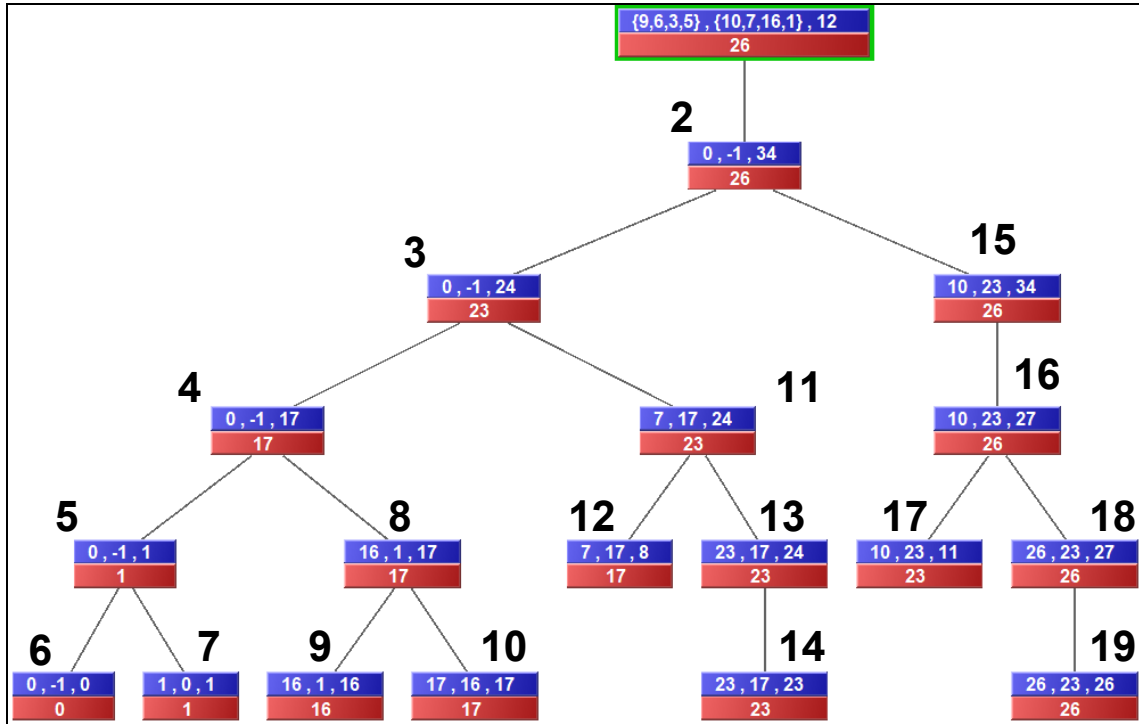
```

El árbol de recursión generado es:

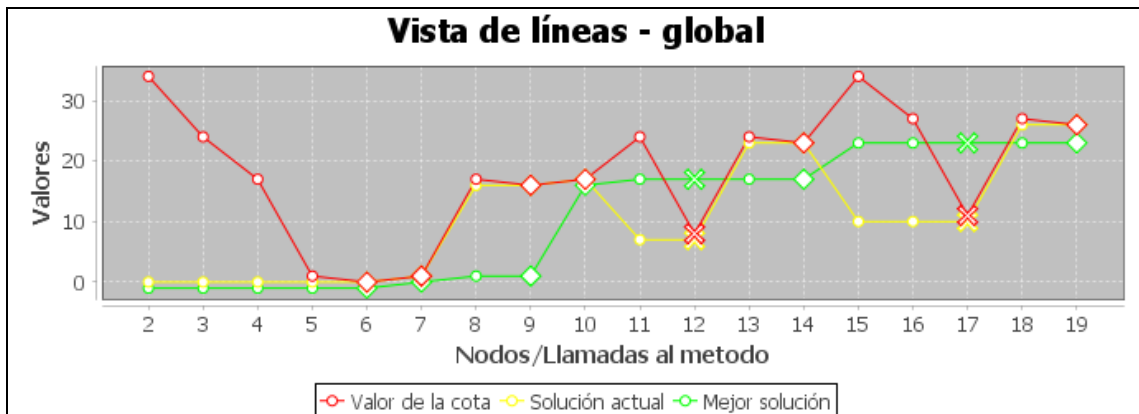


Por favor, contesta a las preguntas del apartado 4 del cuestionario. Si quieres, puedes probar primero SRec con estos mismos datos o con datos tuyos para familiarizarte mejor con ambos algoritmos y las visualizaciones que generan.

5. **Evolución de valores.** Sea de nuevo la última visualización, con sus nodos numerados en orden de su primera visita (en profundidad y preorden):



La siguiente visualización presenta la evolución de los tres valores mostrados en cada nodo a lo largo del recorrido en profundidad. Cada rombo grande representa un nodo de hoja del árbol de recursión, es decir, una solución completa. Asimismo, cada cruz representa un nodo que se poda debido a incumplir la condición de cota.



Por favor, contesta a las preguntas del apartado 5 del cuestionario. Si quieres, puedes probar SRec con estos mismos datos o con datos tuyos para familiarizarte mejor con ambos algoritmos y las visualizaciones que generan.

- 6. Visualizaciones propias.** Si el alumno lo desea, puede generar visualizaciones de sus prácticas 3a y 3b siguiendo los esquemas de programación correspondientes a los algoritmos mostrados en esta sesión. Pueden enviarse dichas visualizaciones en un fichero PDF por medio del apartado de Evaluación del aula virtual. Si se tienen dificultades, puede enviarse por el correo del aula virtual con el asunto “Sesión de SRec”.

Apéndice B: Cuestionario de Opinión de la Primera Evaluación

**Grados en Ingeniería Informática e Ingeniería de Computadores
Asignatura *Algoritmos Avanzados*
Curso 2022/2023
Sesión de familiarización con SRec – Cuestionario**

El objetivo del cuestionario es mejorar el uso de visualizaciones en la asignatura en cursos futuros. La participación es voluntaria.

Nombre y apellidos (opcional): _____

3. Visualización de algoritmos de vuelta atrás.

- Indica con una X si has probado en SRec los algoritmos de vuelta atrás proporcionados con los datos del ejemplo:

SÍ	<input type="checkbox"/>
NO	<input type="checkbox"/>

- Enumera los sucesivos valores que va tomando la variable $bOpt$ hasta llegar al resultado optimal, incluido:

- ¿En cuántos nodos se descarta el candidato $k=1$ porque desbordaría la capacidad de la mochila?

- Indica con una X si has probado en SRec los algoritmos de vuelta atrás proporcionados con otros datos:

SÍ	<input type="checkbox"/>
NO	<input type="checkbox"/>

Si es así, escríbelos:.....

- Valora de 1 a 5 la utilidad de cada visualización para comprender el proceso de búsqueda por vuelta atrás:

	1 = muy poco	2 = poco	3 = regular	4 = bastante	5 = mucho
Árbol 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Árbol 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Valora de 1 a 5 la claridad de cada algoritmo de vuelta atrás:

	1 = muy poco	2 = poco	3 = regular	4 = bastante	5 = mucho
Algoritmo 1					
Algoritmo 2					

- Otros comentarios:
-
-

Grados en Ingeniería Informática e Ingeniería de Computadores
Asignatura Algoritmos Avanzados
Curso 2022/2023
Sesión de familiarización con SRec – Cuestionario

El objetivo del cuestionario es mejorar el uso de visualizaciones en la asignatura en cursos futuros. La participación es voluntaria.

Nombre y apellidos (opcional): _____

4. Visualización de algoritmos de ramificación y poda.

- Indica con una X si has probado en SRec los algoritmos de ramificación y poda proporcionados con los datos del ejemplo:

SÍ	
NO	

- Enumera los sucesivos valores que va tomando la variable *bOpt* hasta llegar al resultado optimal, incluido:

- ¿En cuántos casos se ha producido una poda gracias a la cota? Para cada caso, indica los valores de la cota y de la mejor solución actual:

- Indica con una X si has probado en SRec los algoritmos de ramificación y poda proporcionados con otros datos:

SÍ	
NO	

Si es así, escríbelos:.....

- Valora de 1 a 5 la utilidad de cada visualización para comprender el proceso de búsqueda por ramificación y poda:

	1 = muy poco	2 = poco	3 = regular	4 = bastante	5 = mucho
Árbol 1					
Árbol 2					

- Valora de 1 a 5 la claridad de cada algoritmo de ramificación y poda:

	1 = muy poco	2 = poco	3 = regular	4 = bastante	5 = mucho
Algoritmo 1					
Algoritmo 2					

- Otros comentarios:
.....
.....

Grados en Ingeniería Informática e Ingeniería de Computadores
Asignatura Algoritmos Avanzados
Curso 2022/2023
Sesión de familiarización con SRec – Cuestionario

El objetivo del cuestionario es mejorar el uso de visualizaciones en la asignatura en cursos futuros. La participación es voluntaria.

Nombre y apellidos (opcional): _____

5. Evolución de valores.

- Indica con una X si has probado en SRec las vistas de líneas con los algoritmos de ramificación y poda proporcionados y los datos del ejemplo:

SÍ	
NO	

- ¿Encuentras correspondencia entre los nodos de hoja del árbol de recursión y los nodos de la vista de líneas con un rombo grande?

- ¿Encuentras correspondencia entre los nodos de hoja podados por cota en el árbol de recursión y los nodos de la vista de líneas con una cruz?

- Indica con una X si has probado en SRec los algoritmos de ramificación y poda proporcionados con otros datos:

SÍ	
NO	

Si es así, escríbelos:

- Valora de 1 a 5 la utilidad del uso combinado de árboles de recursión y de vistas de líneas para comprender el proceso de búsqueda:

	1 = muy poco	2 = poco	3 = regular	4 = bastante	5 = mucho
Árbol 1					
Árbol 2					

- Indica qué visualización te parece más útil:

Árboles de recursión	
Vistas de líneas	
Uso combinado de ambas vistas	

- Otros comentarios:
-
-

Apéndice C: Respuestas al Cuestionario de la Primera Evaluación

CUESTIONARIO DE OPINION SOBRE LA VISUALIZACIÓN DE ALGORITMOS DE BÚSQUEDA - CURSO 2022-23

ALUMNO	Alumno 1	Alumno 2	Alumno 3	Alumno 4	Alumno 5	Alumno 6	Alumno 7
Visualización de algoritmos de vuelta atrás							
Indica si has probado los algoritmos con los datos del ejemplo	SÍ	SÍ	SÍ	SÍ	SÍ	SÍ	SÍ
Enumera los valores que va tomando bOpt hasta llegar al optimal	1, 17, 23, 26	1, 17, 23, 26	1, 17, 23, 26	1, 17, 23, 26	-1, 1, 17, 23	-1, 1, 17, 23	-1, 1, 17, 23
¿En cuántos nodos se descarta el candidato porque desbordaría la mochila?	4	4	4	4	1	1	1
Indica si has probado los algoritmos con otros datos	NO	NO	NO	NO	NO	NO	NO
Si es así, escríbelos							
Valora la utilidad de cada visualización para comprender el proceso de búsqueda							
Árbol 1	3	3	3	3	4	5	5
Árbol 2	3	4	2	3	4	5	5
Valora la claridad de cada algoritmo de vuelta atrás							
Algoritmo 1	4	3	4	4	3	4	5
Algoritmo 2	4	3	3	4	3	4	5
Otros comentarios							
Visualización de algoritmos de ramificación y poda							
Indica si has probado los algoritmos con los datos del ejemplo	SÍ				SÍ	SÍ	SÍ
6 casos se poda el árbol							
¿En cuántos casos se ha producido una poda gracias a la cota? Indica los valores de la cota y de la mejor solución actual	{0,-1}, {1,0}, {16,1}, {17,16}, {27,23}, {24,17}			6 casos se poda {0,-1}, {1,0}, {16,1}, {17,16}, {27,23}, {24,17}	1:34, 2:27, 2:24	En 2 casos: El 1º cota=24 y mejor sol. actual 17 El 2º cota=27 y m.s.a. 23	2 veces con cota 24 y mejor sol. 17 y con cota 27 y mejor sol. 23
¿En cuántos nodos se descarta el candidato porque desbordaría la mochila?							
Indica si has probado los algoritmos con otros datos	NO			NO	NO	NO	NO
Si es así, escríbelos							
Valora la utilidad de cada visualización para comprender el proceso de búsqueda							
Árbol 1	3			3	4	5	4
Árbol 2	3			3	4	5	4
Valora la claridad de cada algoritmo de ramificación y poda							
Algoritmo 1	4			3	3	5	5
Algoritmo 2	4			3	3	5	5
Otros comentarios							
Evolución de valores							
Indica si has probado las vistas de líneas con los algoritmos y datos del ejemplo	NO			NO	SÍ	SÍ	SÍ
¿Encuentras correspondencia entre los nodos de hoja y los nodos de la vista de líneas con un rombo grande?						SÍ, correctamente	SÍ
¿Encuentras correspondencia entre los nodos podados por cota y los nodos de la vista de líneas con una cruz?						SÍ	SÍ
Indica si has probado las vistas de líneas con otros datos	NO			NO	NO	NO	NO
Si es así, escríbelos							
Valora la utilidad del uso combinado de árboles de recursión y de vistas de líneas para comprender el proceso de búsqueda							
Árbol 1					4	4	4
Árbol 2					4	4	4
Indica qué visualización te parece más útil					Uso combinado	Uso combinado	Uso combinado

CUESTIONARIO DE OPINION SOBRE LA VISUALIZACIÓN DE ALGORITMOS DE BÚSQUEDA - CURSO 2022-23

ALUMNO	Alumno 8	Alumno 9	Alumno 10	Alumno 11
Visualización de algoritmos de vuelta atrás				
Indica si has probado los algoritmos con los datos del ejemplo	SÍ	SÍ	SÍ	SÍ
Enumera los valores que va tomando bOpt hasta llegar al optimal	1, 17, 23, 26	1, 17, 23, 26	-1, 1, 17, 23	1, 17, 23, 26
¿En cuántos nodos se descarta el candidato porque desbordaría la mochila?	4	1	7	2
Indica si has probado los algoritmos con otros datos	SÍ	NO	NO	SÍ
Si es así, escríbelos	{9,6,5,3} {10,16,7,1} 14 bOpt=24			{8,16,16,32,64} {1,1,1,1,1} 64
Valora la utilidad de cada visualización para comprender el proceso de búsqueda				
Árbol 1	4	3	5	5
Árbol 2	4	3	5	5
Valora la claridad de cada algoritmo de vuelta atrás				
Algoritmo 1	5	4	4	3
Algoritmo 2	5	4	4	3
Otros comentarios				
Visualización de algoritmos de ramificación y poda				
Indica si has probado los algoritmos con los datos del ejemplo	SÍ	SÍ	SÍ	SÍ
¿En cuántos casos se ha producido una poda gracias a la cota? Indica los valores de la cota y de la mejor solución actual	En 3 casos Cota -> 16, 23, 26 bOpt -> 1, 17, 23 respectivamente	2 {17,24}, {23,27}	2 casos 1º caso: cota=23 solactual=17 2º caso: cota=26 solactual=23	5, <17,24>, <17,24>, <27,27>, <23,27>, <23,34>
¿En cuántos nodos se descarta el candidato porque desbordaría la mochila?				
Indica si has probado los algoritmos con otros datos	SÍ	NO	NO	NO
Si es así, escríbelos				
Valora la utilidad de cada visualización para comprender el proceso de búsqueda				
Árbol 1	4	3	5	4
Árbol 2	4	3	5	4
Valora la claridad de cada algoritmo de ramificación y poda				
Algoritmo 1	5	5	5	3
Algoritmo 2	5	5	5	3
Otros comentarios				
Evolución de valores				
Indica si has probado las vistas de líneas con los algoritmos y datos del ejemplo	SÍ		SÍ	SÍ
¿Encuentras correspondencia entre los nodos de hoja y los nodos de la vista de líneas con un rombo grande?	Sí, son los mismos, representan una solución completa		SÍ	No
¿Encuentras correspondencia entre los nodos podados por cota y los nodos de la vista de líneas con una cruz?	Sí, son los mismos		No	No
Indica si has probado las vistas de líneas con otros datos	SÍ		NO	NO
Si es así, escríbelos				
Valora la utilidad del uso combinado de árboles de recursión y de vistas de líneas para comprender el proceso de búsqueda				
Árbol 1	4		5	5
Árbol 2	4		5	5
Indica qué visualización te parece más útil	Uso combinado		Árboles de recursión	Vistas de líneas
Otros comentarios				Ahora la cabecera tiene sentido para mí

Apéndice D: Enunciado de la Segunda Sesión de Familiarización

Grados en Ingeniería Informática e Ingeniería de Computadores Asignatura *Algoritmos Avanzados* Curso 2023/2024 Sesión de familiarización con SRec

El objetivo de la sesión es familiarizarse con el uso del sistema de visualización de la recursividad SRec, disponible en el aula virtual y que se utilizará en prácticas posteriores.

Como parte de la sesión, se pide responder un cuestionario con tres partes cortas, de forma voluntaria. Las respuestas recogidas servirán para mejorar la asignatura en cursos futuros.

Material

Se dispone de dos clases de Java, *Varios.java* y *Mochila01.java*.

Guion

La sesión consiste en la realización individual de cinco actividades, todas ellas voluntarias. Las dos primeras sirven de familiarización con SRec, y se basan en el fichero *Varios.java*. Las otras tres actividades buscan valorar las visualizaciones generadas para algoritmos de vuelta atrás y de ramificación y poda, y se basan en el fichero *Mochila_0_1.java*.

1. **Iniciación.** SRec es una aplicación JAR. Debe instalarse en una unidad y directorio con permisos de escritura, ya que SRec crea ficheros temporales.

La primera vez que se ejecuta SRec, debe indicarse un directorio donde se encuentre un fichero “java.exe” o “javac.exe” contenido en el directorio “bin” de un JDK de Java (no sirve JRE) instalado en su ordenador (probablemente en el directorio “Archivos de programa”). Al menos se ha probado con los JDK v.7 y v.8.

Sean los dos métodos siguientes para el problema de las torres de Hanoi. En conjunto, calculan el total de movimientos de discos a realizar, imprimiéndolos en el terminal:

```
public static int hanoi (int n) {
    System.out.println
        ("Van a moverse " + n + " discos");
    int mov = h (n, 'A', 'B', 'C');
    System.out.println
        ("Se han realizado " + mov + " movimientos");
    return mov;
}

private static int h (int n, char a, char b, char c) {
    int mov = 0;
    if (n!=0) {
        int mov1 = h (n-1, a,c,b);
        System.out.println ("se mueve un disco del poste "
            + a + " al poste " + c);
        int mov2 = h (n-1, b,a,c);
        mov = mov1 + 1 + mov2;
    }
    return mov;
}
```

Se pide usar las principales funciones de SRec:

- Cargar el fichero Java *varios.java*.
- Seleccionar la función *hanoi* y lanzar una o varias ejecuciones con valores de *n* comprendidos entre 3 y 6.
- Experimentar con los controles de animación, observando su efecto sobre las vistas de la recursividad que ofrece SRec:
 - o Traza.
 - o Pila de control.
 - o Árbol de activación. Si el árbol es muy grande, sólo se muestra una parte de la vista, pero se facilita la navegación por el mismo mediante un visor situado en la parte inferior de la vista del árbol.
 - o Terminal.
- Interacción con las vistas:
 - o Mover los paneles e incluso ocultarlos.
 - o Borrar o cerrar el terminal.

- 2. Interacción.** Sean los siguientes métodos que calculan la suma de los elementos de un vector mediante la técnica de divide y vencerás:

```
public static int sumaArray (int[] v) {
    return sumaDyV (v, 0, v.length-1);
}

private static int sumaDyV (int[] v,
                             int inf, int sup) {

    if (inf==sup)
        return v[inf];
    else {
        int med = (inf+sup)/2;
        int suma1 = sumaDyV (v, inf, med);
        int suma2 = sumaDyV (v, med+1, sup);
        return suma1 + suma2;
    }
}
```

Se pide generar una animación cualquiera y usar las principales funciones de SRec de filtrado y configuración:

- Control de la cantidad de información mostrada en el árbol (métodos, entrada/salida, parámetros).
- Control del zoom.
- Control del formato gráfico de las 3 visualizaciones (traza, pila de control, árbol de activación).
- Cambiar las vistas de un panel a otro.
- Almacenar una o varias vistas en ficheros gráficos (pueden contemplarse con el visor de imágenes de Windows).

NOTA: El cuestionario a rellenar (de forma voluntaria) se refiere a las partes siguientes. Puedes rellenarlo en papel en caso de encontrarte en el aula informática o rellenarlo electrónicamente y enviarlo como fichero PDF por medio del apartado de Evaluación del aula virtual. Si se tienen dificultades, puede enviarse por el correo del aula virtual con el asunto “Sesión de SRec”.

3. **Visualización de algoritmos de vuelta atrás.** Sea el siguiente algoritmo desarrollado para resolver el conocido *problema de la mochila 0/1* conforme a uno de los esquemas de codificación sugerido para la técnica de vuelta atrás, que tiene la estructura usual de los algoritmos recursivos:

```

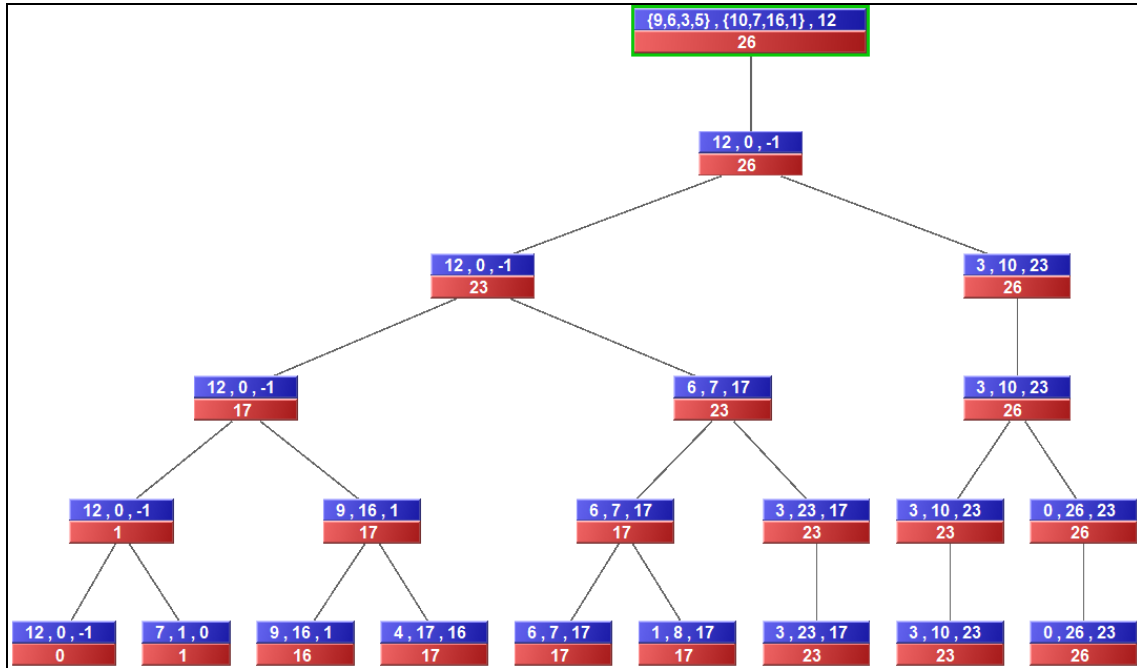
public static int mochila01_back
    (int[]ps, int[]bs, int c) {
    int[] solParcial = new int[ps.length];
    int[] solOptima = new int[ps.length];
    int bOpt =
        buscar01a(ps,bs,c,0,c,solParcial,0,solOptima,-1);
    return bOpt;
}

private static int buscar01a(int[] ps, int[] bs, int c,
    int i, int p,
    int[] solParc, int b,
    int[] solOpt, int bOpt) {

    if (i==ps.length) {
        if (b>bOpt) {
            bOpt = b;
            for (int j=0; j<ps.length; j++)
                solOpt[j] = solParc[j];
        }
    } else
        for (int k=0; k<=1; k++) {
            if (k*ps[i]<=p) {
                solParc[i] = k;
                int np = p - k*ps[i];
                int nb = b + k*bs[i];
                bOpt = buscar01a
                    (ps,bs,c,i+1,np,solParc,nb,solOpt,bOpt);
            }
        }
    return bOpt;
}

```

que genera el siguiente árbol de recursión:



Antes de continuar, dedica algo de tiempo a comprender la figura. Si quieres, puedes probar SRec con estos mismos datos o con datos tuyos para familiarizarte mejor con el algoritmo y con las visualizaciones que genera.

Por favor, contesta a las preguntas del apartado 3 del cuestionario.

4. **Visualización de algoritmos de ramificación y poda.** Sea ahora el siguiente algoritmo de ramificación y poda, cuya función de cota es igual a la suma de los beneficios. La comprobación de cota es la primera instrucción de cada llamada recursiva.

```

public static int mochila01_RyP
    (int[] ps, int[] bs, int c) {
    int[] solParcial = new int[ps.length];
    int[] solOptima = new int[ps.length];
    // cota como suma de beneficios
    int cota = 0;
    for (int i=0; i<bs.length; i++)
        cota += bs[i];
    int bOpt = buscar01b
        (ps,bs,c,0,c,solParcial,0,solOptima,-1,cota);
    return bOpt;
}

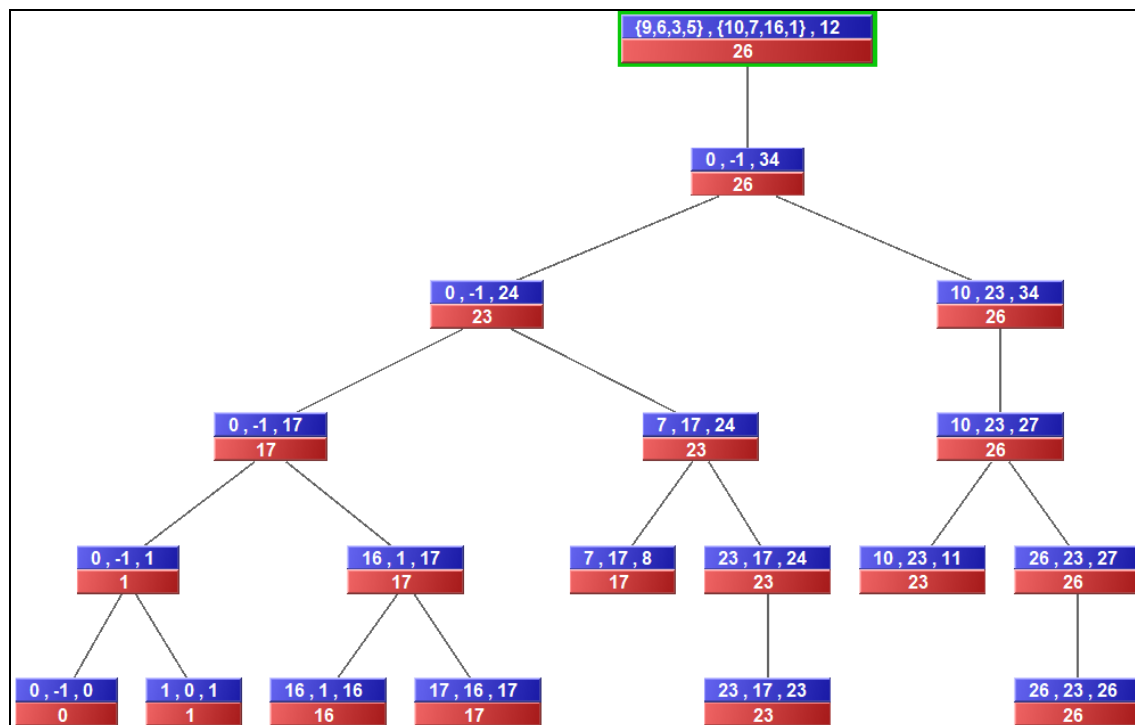
private static int buscar01b(int[] ps, int[] bs, int c,
    int i, int p,
    int[] solParc, int b,
    int[] solOpt, int bOpt,
    int cota) {

    if (cota>bOpt)
        if (i==ps.length) {
            if (b>bOpt) {
                bOpt = b;
                for (int j=0; j<ps.length; j++)
                    solOpt[j] = solParc[j];
            }
        } else
            for (int k=0; k<=1; k++) {
                if (k*ps[i]<=p) {
                    solParc[i] = k;
                    int np = p - k*ps[i];
                    int nb = b + k*bs[i];
                    int nCota = cota - (1-k)*bs[i];
                    bOpt = buscar01b
                        (ps,bs,c,i+1,np,solParc,nb,solOpt,bOpt,nCota);
                }
            }
    return bOpt;
}

```

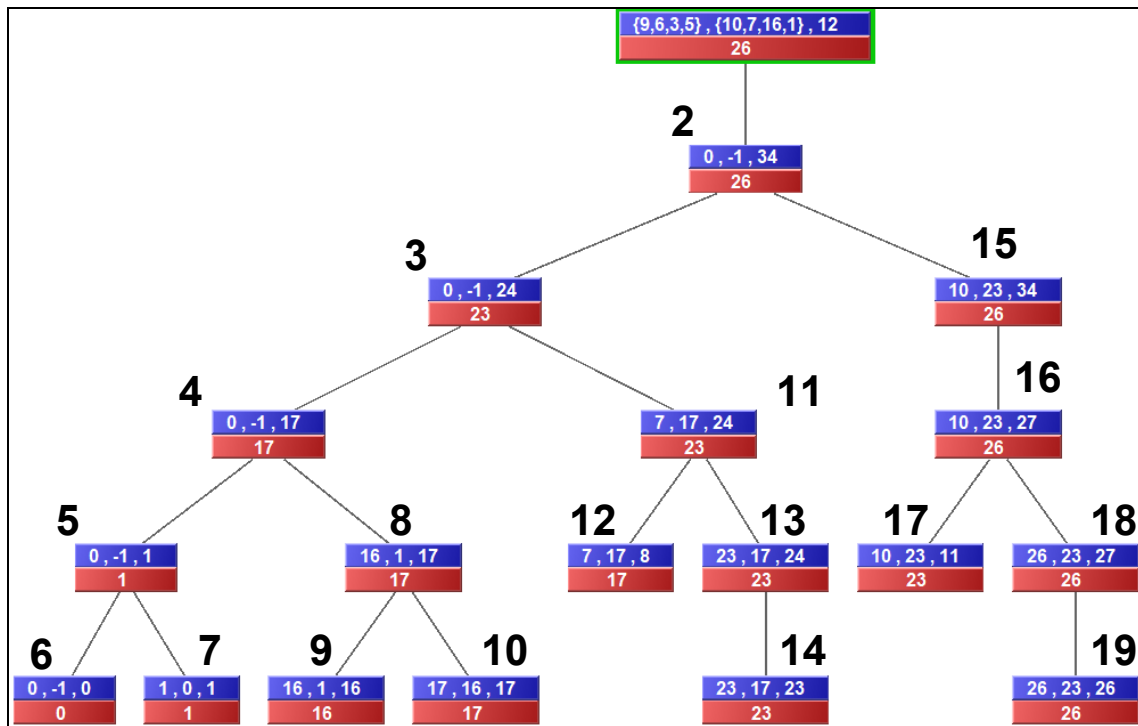
El árbol de recursión generado es el siguiente.

De nuevo, dedica algo de tiempo a comprender la figura antes de continuar. Si quieres, puedes probar primero SRec con estos mismos datos o con datos tuyos para familiarizarte mejor con el algoritmo y las visualizaciones que genera.



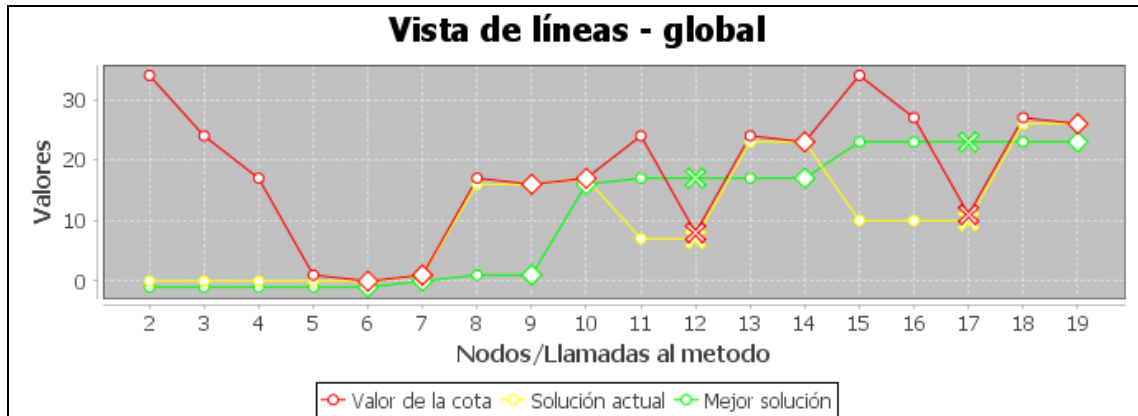
Por favor, contesta a las preguntas del apartado 4 del cuestionario.

5. **Evolución de valores durante el proceso de búsqueda.** Sea la última visualización, con sus nodos numerados en orden de su primera visita (en profundidad y preorden):



La siguiente visualización presenta la evolución de los tres valores mostrados en cada nodo del árbol a lo largo de su recorrido en profundidad. Se ha obtenido mediante la vista de líneas, que debe activarse al procesar el fichero para todos los métodos que se quiere visualizar con esta vista (en este caso, el método buscar01b).

La vista de líneas muestra la evolución temporal de tres valores según explora el algoritmo los sucesivos nodos del árbol de búsqueda: valor de la cota, valor de la solución en construcción y valor de la mejor solución encontrada hasta el momento. Cada rombo grande representa un nodo de hoja del árbol de búsqueda, es decir, una solución completa. Asimismo, cada cruz representa un nodo que se poda debido a incumplir la condición de cota.



De nuevo, dedica algo de tiempo a comprender la figura antes de continuar. Si quieres, puedes probar SRec con estos mismos datos o con datos tuyos para familiarizarte mejor con esta visualización del algoritmo de ramificación y poda.

Por favor, contesta a las preguntas del apartado 5 del cuestionario.

- 6. Visualizaciones de algoritmos propios.** Si lo deseas, puedes generar visualizaciones de tu práctica 3 siguiendo los esquemas de programación correspondientes a los algoritmos mostrados en esta sesión. Pueden enviarse dichas visualizaciones en un fichero PDF por medio del apartado de Evaluación del aula virtual. Si se tienen dificultades, puede enviarse por el correo del aula virtual con el asunto “Sesión de SRec”.

Apéndice E: Cuestionario de Opinión de la Segunda Evaluación

**Grados en Ingeniería Informática e Ingeniería de Computadores
Asignatura *Algoritmos Avanzados*
Curso 2023/2024
Sesión de familiarización con SRec – Cuestionario**

El objetivo del cuestionario es mejorar el uso de visualizaciones en la asignatura en cursos futuros. La participación es voluntaria.

Nombre y apellidos (opcional): _____

3. Visualización de algoritmos de vuelta atrás.

- Indica con una X si has probado en SRec el algoritmo de vuelta atrás proporcionado con los datos del ejemplo:

SÍ	<input type="checkbox"/>
NO	<input type="checkbox"/>

- Enumera los sucesivos valores que va tomando la variable b_{Opt} hasta llegar al resultado optimal, incluido:

- ¿En cuántos nodos se descarta el candidato $k=1$ porque desbordaría la capacidad de la mochila?

- Indica con una X si has probado en SRec el algoritmo de vuelta atrás proporcionado con otros datos:

SÍ	<input type="checkbox"/>
NO	<input type="checkbox"/>

Si es así, escríbelos:.....

- Valora de 1 a 5 la utilidad de la visualización para comprender el proceso de búsqueda por vuelta atrás:

1 = muy poco	2 = poco	3 = regular	4 = bastante	5 = mucho
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Valora de 1 a 5 la claridad del código del algoritmo de vuelta atrás:

1 = muy poco	2 = poco	3 = regular	4 = bastante	5 = mucho

- Otros comentarios:
-
-

Grados en Ingeniería Informática e Ingeniería de Computadores
Asignatura Algoritmos Avanzados
Curso 2023/2024
Sesión de familiarización con SRec – Cuestionario

El objetivo del cuestionario es mejorar el uso de visualizaciones en la asignatura en cursos futuros. La participación es voluntaria.

Nombre y apellidos (opcional): _____

4. Visualización de algoritmos de ramificación y poda.

- Indica con una X si has probado en SRec el algoritmo de ramificación y poda proporcionado con los datos del ejemplo:

SÍ	
NO	

- ¿En cuántos casos se ha producido una poda gracias a la cota? Para cada caso, indica los valores de la cota y de la mejor solución actual:

- Indica con una X si has probado en SRec el algoritmo de ramificación y poda proporcionado con otros datos:

SÍ	
NO	

Si es así, escríbelos:.....

- Valora de 1 a 5 la utilidad de la visualización para comprender el proceso de búsqueda por ramificación y poda:

1 = muy poco	2 = poco	3 = regular	4 = bastante	5 = mucho

- Valora de 1 a 5 la claridad del código del algoritmo de ramificación y poda:

1 = muy poco	2 = poco	3 = regular	4 = bastante	5 = mucho

- Otros comentarios:
.....
.....

Grados en Ingeniería Informática e Ingeniería de Computadores
Asignatura Algoritmos Avanzados
Curso 2023/2024
Sesión de familiarización con SRec – Cuestionario

El objetivo del cuestionario es mejorar el uso de visualizaciones en la asignatura en cursos futuros. La participación es voluntaria.

Nombre y apellidos (opcional): _____

5. Evolución de valores durante el proceso de búsqueda.

- Indica con una X si has probado en SRec las vistas de líneas con el algoritmo de ramificación y poda proporcionado y los datos del ejemplo:

SÍ	<input type="checkbox"/>
NO	<input type="checkbox"/>

- ¿Encuentras correspondencia entre los nodos de hoja del árbol de recursión y los nodos de la vista de líneas con un rombo grande?

- ¿Encuentras correspondencia entre los nodos de hoja podados por cota en el árbol de recursión y los nodos de la vista de líneas con una cruz?

- Indica con una X si has probado en SRec las vistas de líneas con el algoritmo de ramificación y poda proporcionado usando otros datos:

SÍ	<input type="checkbox"/>
NO	<input type="checkbox"/>

Si es así, escríbelos:.....

- Valora de 1 a 5 la utilidad del uso combinado de árboles de recursión y de vistas de líneas para comprender el proceso de búsqueda:

1 = muy poco	2 = poco	3 = regular	4 = bastante	5 = mucho
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Indica qué visualización te parece más útil:

Árboles de recursión	
Vistas de líneas	
Uso combinado de ambas vistas	

- Otros comentarios:
-
-

Apéndice F: Respuestas al Cuestionario de la Segunda Evaluación

CUESTIONARIO DE OPINION SOBRE LA VISUALIZACIÓN DE ALGORITMOS DE BÚSQUEDA - CURSO 2023-24

ALUMNO	Participante 1	Participante 2	Participante 3	Participante 4	Participante 5	Participante 6
Visualización de algoritmos de vuelta atrás						
Indica si has probado los algoritmos con los datos del ejemplo	SÍ	SÍ	SÍ	SÍ	SÍ	SÍ
Enumera los valores que va tomando bOpt hasta llegar al optimal	0,1,16,17,23,26	-1,0,1,16,17,23	-1,0,1,16,17,23	-1,23,23,23,23	-1,0,1,16,17,23,26	Resultado final 26 1,16,17,23,26 no sabemos suponiendo k=1, en 2 nodos
¿En cuántos nodos se descarta el candidato porque desbordaría la mochila?	7	7	7	2 {1,1,1,0} y {0,1,1,1}	4	NO
Indica si has probado los algoritmos con otros datos	NO	NO	NO	NO	SÍ {3,2,1,4} {10,10,4,1}	NO
Si es así, escríbelos					12	
Valora la utilidad de la visualización para comprender el proceso de búsqueda	5	4	5	4	4	4
Valora la claridad del algoritmo de vuelta atrás	5	4	4	3	4	5
Otros comentarios						
Visualización de algoritmos de ramificación y poda						
Indica si has probado los algoritmos con los datos del ejemplo	SÍ	SÍ	SÍ	SÍ	SÍ	SÍ
¿En cuántos casos se ha producido una poda gracias a la cota? Indica los valores de la cota y de la mejor solución actual	2 cota: 8 cota: 11 mejor solución cota 8: 0,0,1,1 " " cota 11: 0,1,1,0	2 cota 8 y cota 11 Cota 8: 0,0,1,1. Cota 11 máx: 0,1,1,0.	2 cota: 8 cota: 11 0011 0110	3 para los de la izda cota=8 y mejor solución 17 para el de la drcha, cota=11 y mejor solución 23	Se producen 2 casos Caso 1: cota=8 bOpt=17 Caso 2: cota=11 bOpt=23	2 veces, Cota=8, máximo 17 Cota=11, máximo 23
¿En cuántos nodos se descarta el candidato porque desbordaría la mochila?						
Indica si has probado los algoritmos con otros datos	NO	NO	NO	NO	SÍ {3,2,1,4} {10,10,4,1}	NO
Si es así, escríbelos					12	
Valora la utilidad de la visualización para comprender el proceso de búsqueda	4	4	5	4	5	5
Valora la claridad del algoritmo de ramificación y poda	4	4	4	4	4	5
Otros comentarios						
Evolución de valores						
Indica si has probado las vistas de líneas con los algoritmos y datos del ejemplo	SÍ	SÍ	SÍ	SÍ	SÍ	SÍ
¿Encuentras correspondencia entre los nodos de hoja y los nodos de la vista de líneas con un rombo grande?	SÍ	SÍ	SÍ	No	Donde se encuentra una mejor solución óptima	Sí, todo se entiende perfectamente en el gráfico con el árbol
¿Encuentras correspondencia entre los nodos podados por cota y los nodos de la vista de líneas con una cruz?	SÍ	SÍ	SÍ	Sí, poda las soluciones actuales que no son óptimas	La cota es menor que la mejor solución encontrada hasta el momento	" igual
Indica si has probado las vistas de líneas con otros datos	NO	NO	NO	NO	SÍ {3,2,1,4} {10,10,4,1}	NO
Si es así, escríbelos					12	
Valora la utilidad del uso combinado de árboles de recursión y de vistas de líneas para comprender el proceso de búsqueda	5	4	5	4	5	5
Indica qué visualización te parece más útil	Uso combinado	Uso combinado	Uso combinado	Árboles de recursión	Uso combinado	Uso combinado