Aprendizaje de algoritmos: sistemas interactivos y dificultades de los alumnos

Ángel Velázquez Iturbide
Departamento de Informática y Estadística
Universidad Rey Juan Carlos
Madrid, España

Universidad Rey Juan Carlos

Lite

Índice

1 Introducción

- Importancia de los algoritmos en los planes de estudio de informática
- Pensamiento algorítmico:
  - Diseño y análisis de métodos precisos de resolución de problemas
  - Antecedente del pensamiento computacional

1 Introducción

- Presentación de algunos resultados del objetivo de investigación:
  - Mejorar al aprendizaje de los algoritmos con ayuda del ordenador
- Herramientas informáticas:
  - Visualización (de recursividad):
    - SRec
  - Experimentación (con optimidad):
    - GreedEx
    - OptimEx

2 Sistema SRec

- Importancia de la visualización:
  - Fuerza expresiva (“una imagen vale más que mil palabras”)
- Ejemplos diarios:
  - Mapas
  - Diagramas
  - Dibujos anatómicos
  - ...

2 Sistema SRec

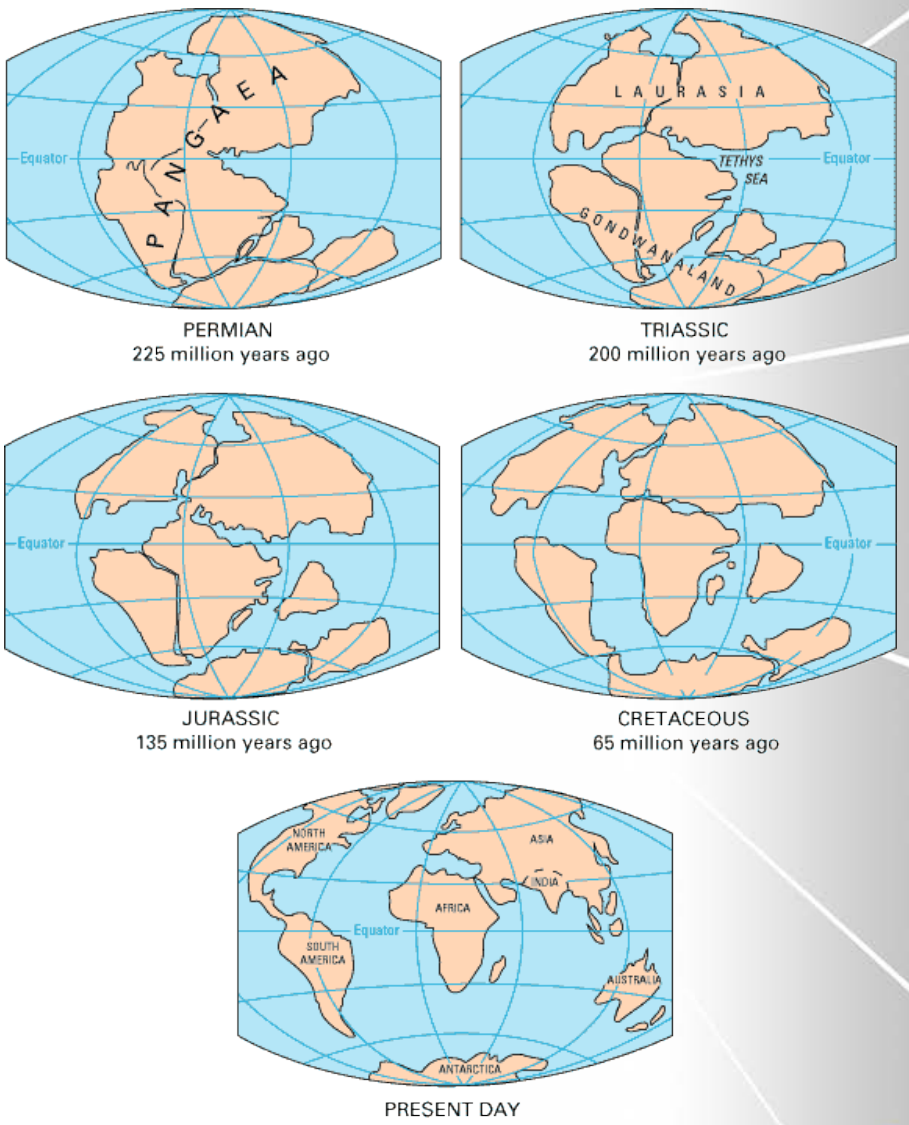
- Visualizar ≠ ver
- Visualizar (María Moliner):
  - "hacer visible mediante aparatos lo que no lo es a simple vista"
  - "representar mentalmente nociones abstractas"
- Ver (María Moliner):
  - "percibir algo por el sentido de la vista"
  - "entender una cosa"

2 Sistema SRec

- La visualización como interpretación:

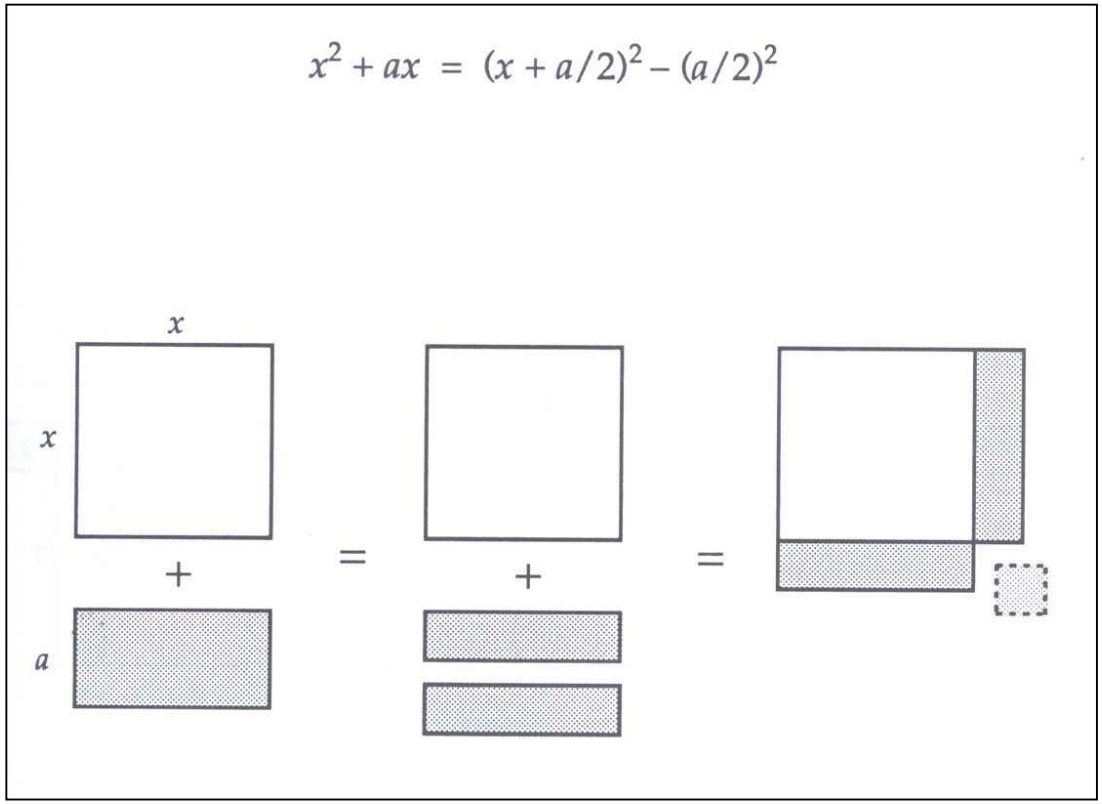
2 Sistema SRec

- La visualización en las explicaciones:



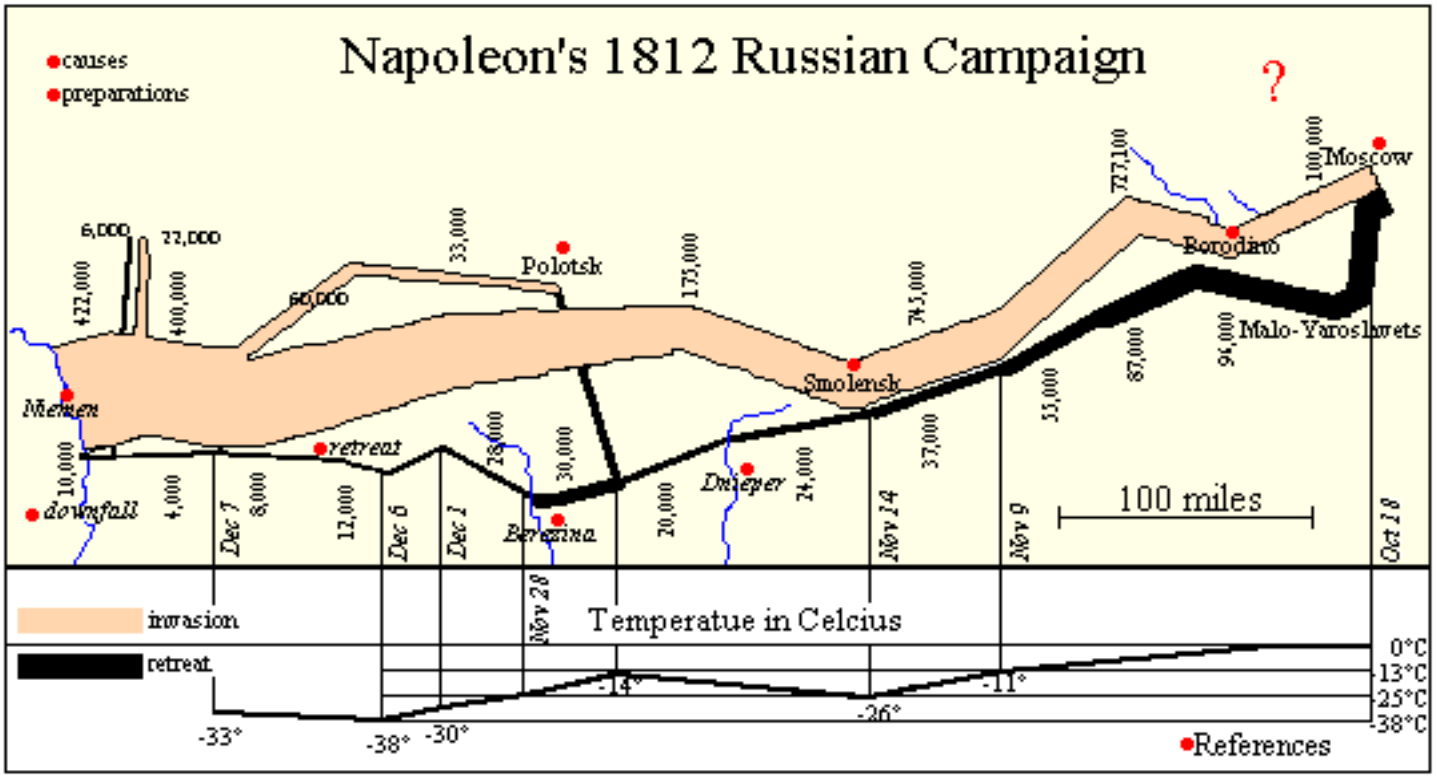
2 Sistema SRec

- La visualización en las explicaciones:



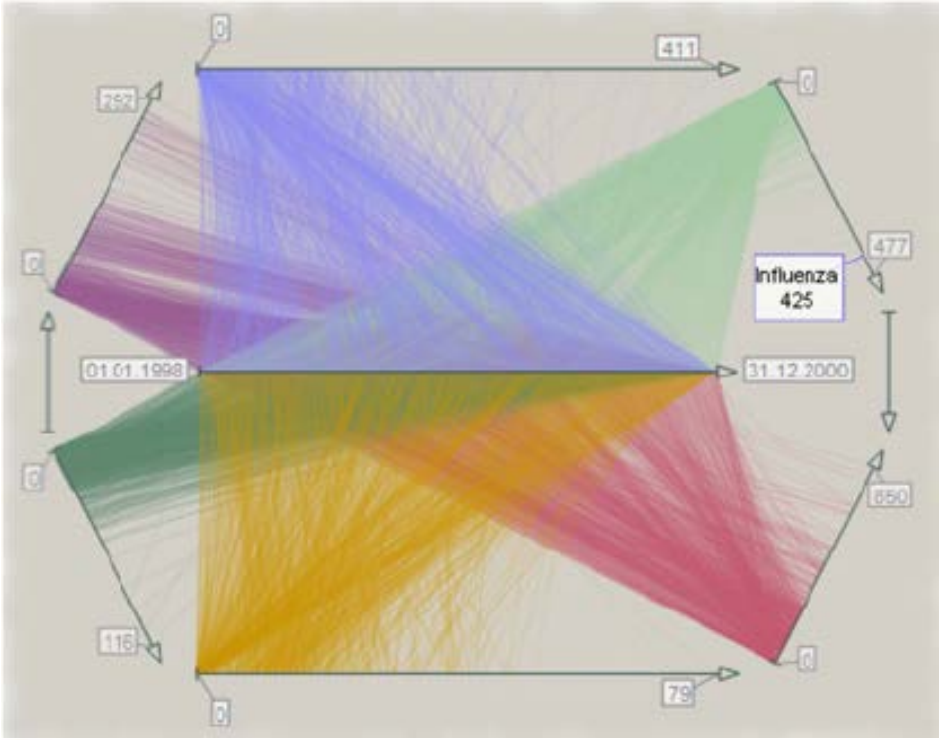
2 Sistema SRec

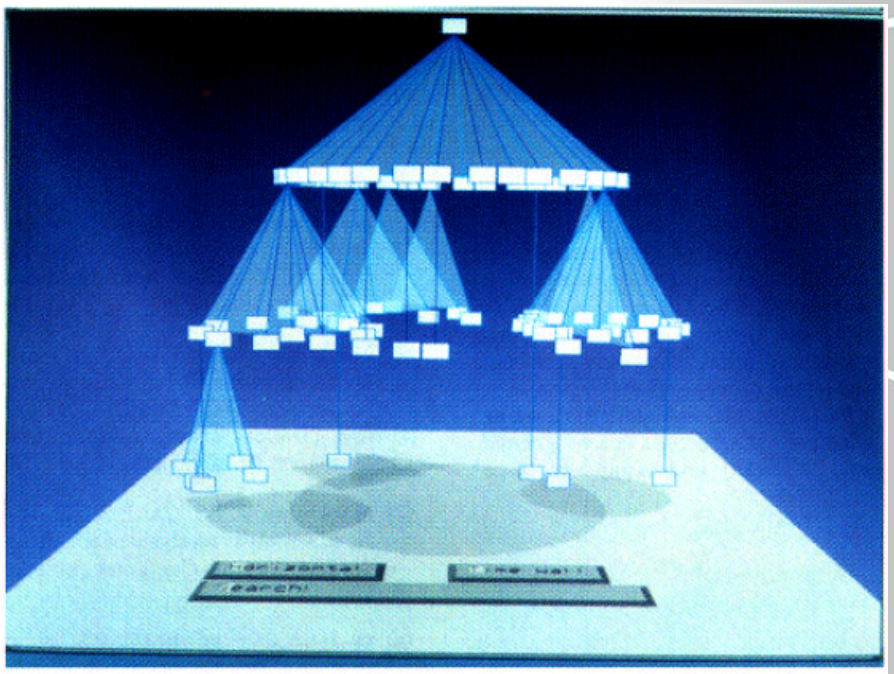
- La visualización en las explicaciones:



2 Sistema SRec

- La visualización de la información:





Robertson Plate 1

2 Sistema SRec

- Uso de la visualización de algoritmos como forma de mejorar:
  - Eficacia
  - Eficiencia
  - Motivación

2 Sistema SRec

- Sistema de visualización de la recursividad:
  - Visualización automática
  - Muy interactivo
- Razones para visualizar la recursividad:
  - Alto nivel de abstracción
  - Mecanismo utilizado en numerosas técnicas de diseño de algoritmos

2 Sistema SRec

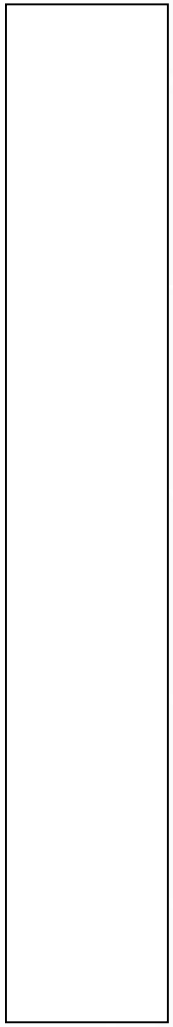
- www.lite.etsii.urjc.es/tools/srec/

Universidad Rey Juan Carlos

Lite

2 Sistema SRec

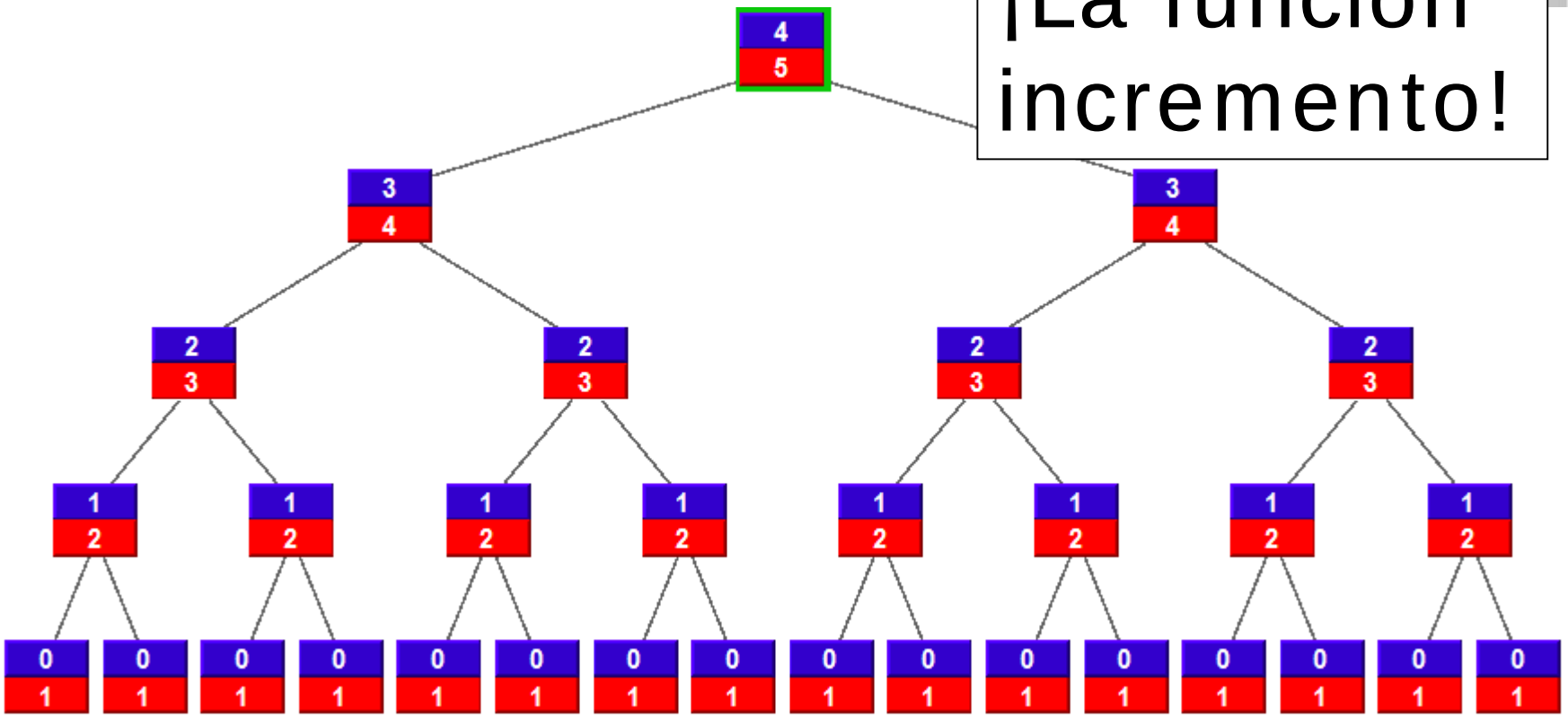
- Usos de SRec:
  - Aprendizaje de la recursividad:



2 Sistema SRec

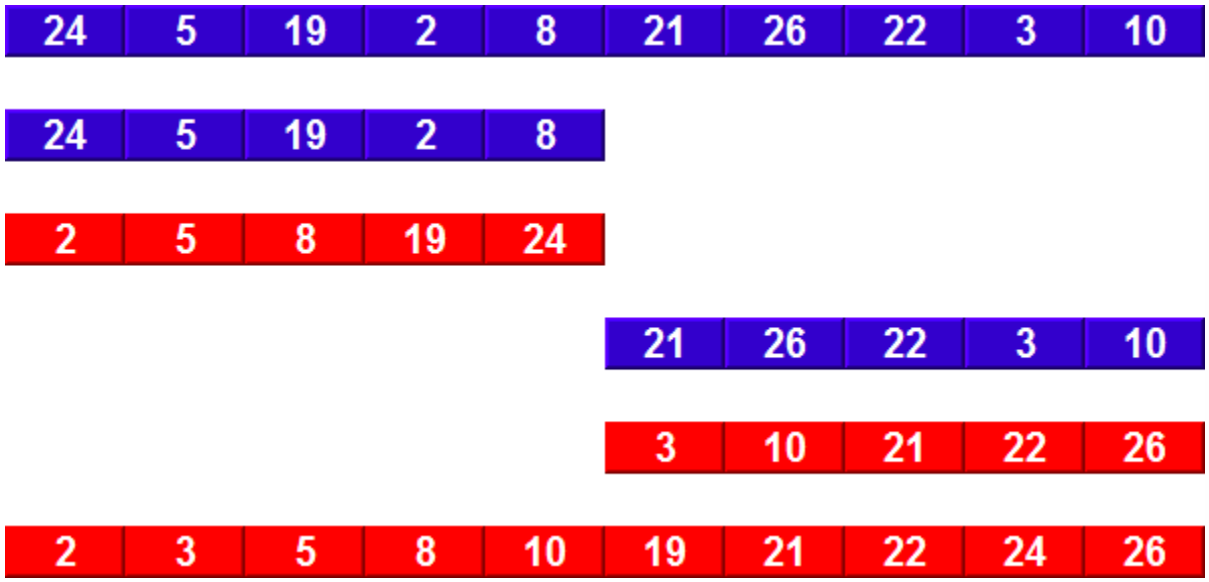
- Usos de SRec:
  - Comprender el comportamiento de un algoritmo recursivo:

¡La función incremento!



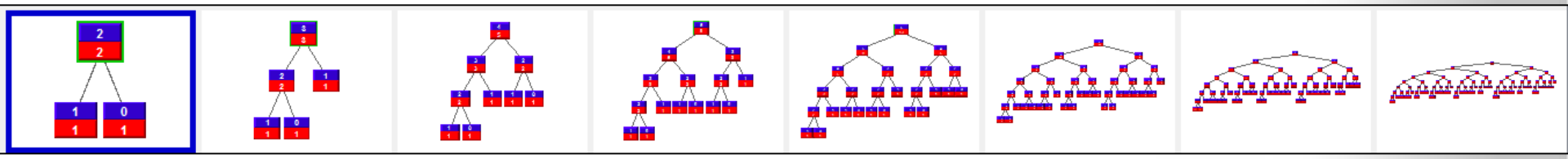
2 Sistema SRec

- Usos de SRec:
  - Comprender el comportamiento de un algoritmo recursivo:
    - Ordenación por mezcla (*mergesort*):



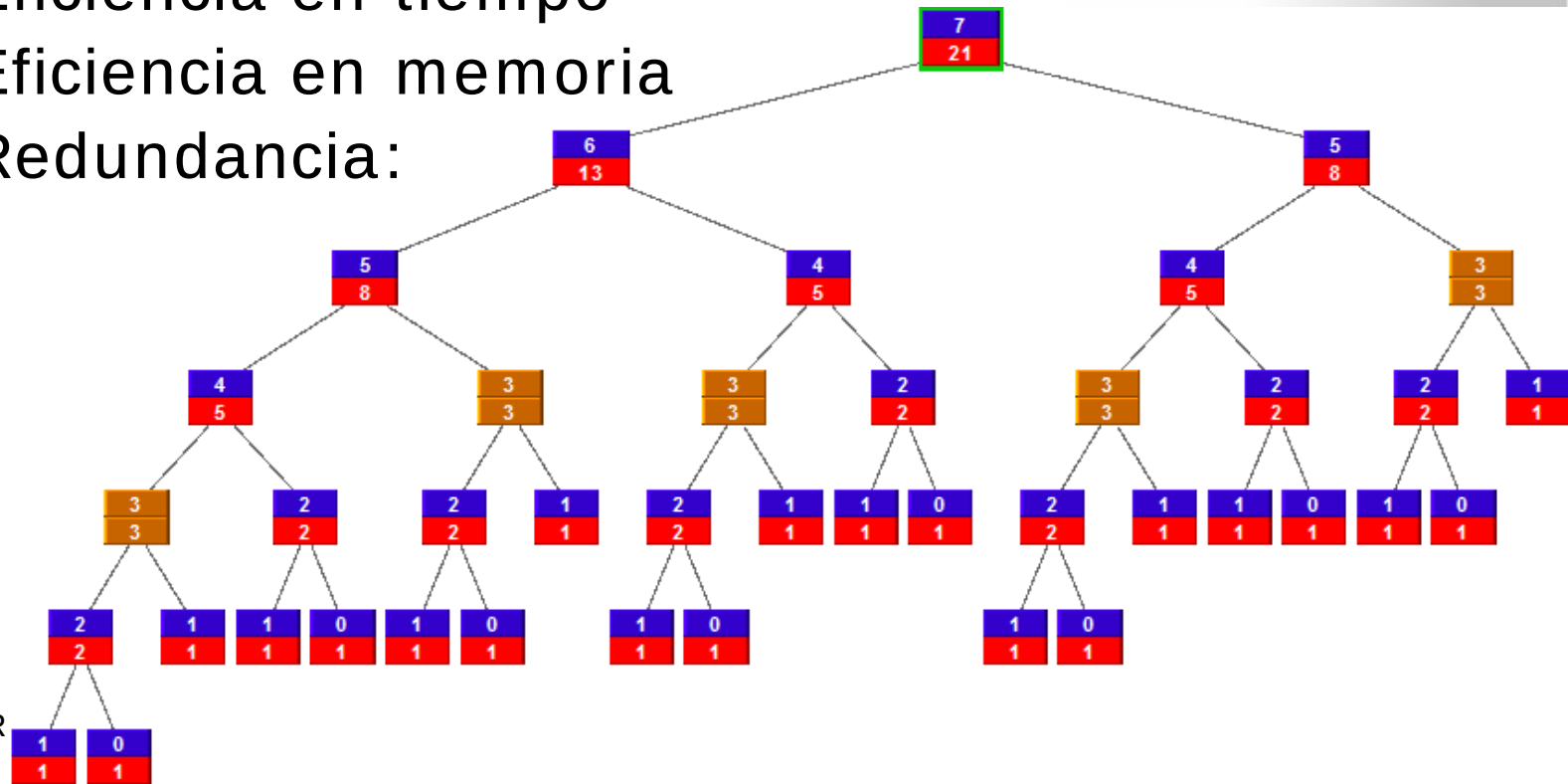
2 Sistema SRec

- Usos de SRec:
  - Análisis de eficiencia de un algoritmo recursivo:
    - Crecimiento asintótico:



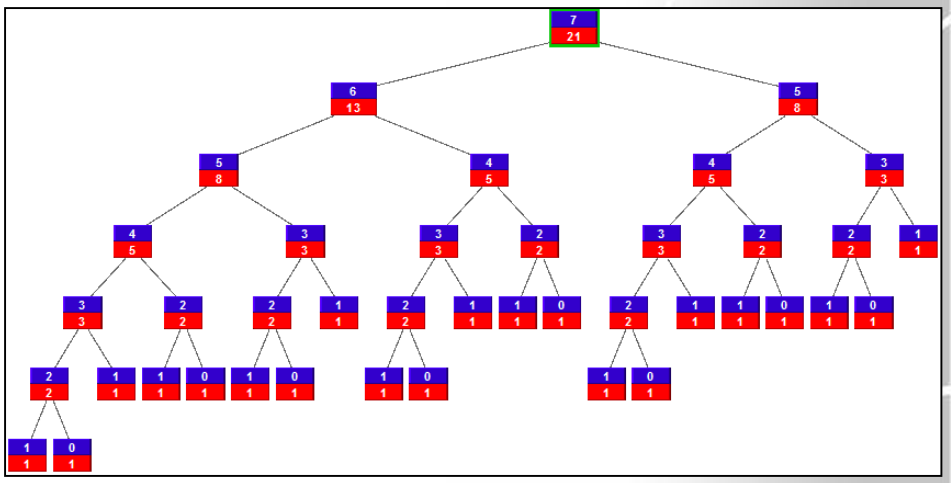
2 Sistema SRec

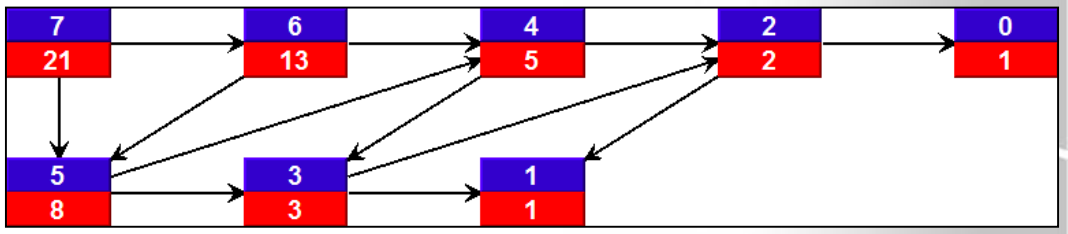
- Usos de SRec:
  - Análisis de eficiencia de un algoritmo recursivo:
    - Eficiencia en tiempo
    - Eficiencia en memoria
    - Redundancia:

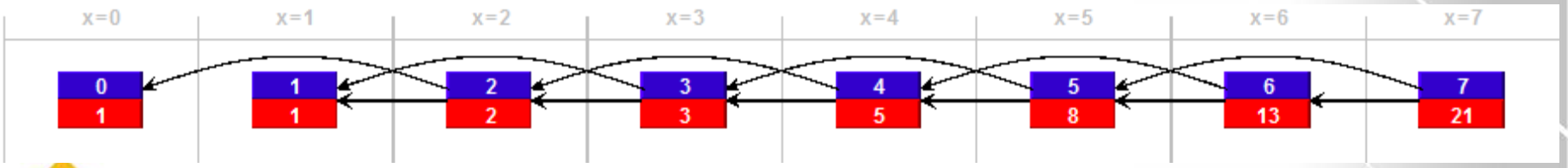


2 Sistema SRec

- Usos de SRec:
  - Conversión de algoritmos redundantes en otros no redundantes:
    - Fibonacci:







3 Sistema GreedEx

- Experimentación como forma de mejorar:
  - Motivación
  - Comprensión de conceptos
    - e identificación de conceptos erróneos
- Integración imprescindible de los sistemas y la didáctica

3 Sistema GreedEx

- Contexto de nuestro trabajo:
  - Problemas de optimización
  - Estudio de las propiedades de los algoritmos:

Propiedades	**Análisis formal**	**Análisis experimental**
Corrección	Verificación	Pruebas
Eficiencia	Complejidad	Medidas
Optimidad	Demostración	∅

3 Sistema GreedEx

- Dificultades para la docencia de los algoritmos voraces:
  - Aparente sencillez, pero con objetivos de aprendizaje exigentes
  - Presentación pasiva de contenidos

3 Sistema GreedEx

- Ejemplo
- Problema de selección de actividades:

i	0	1	2	3	4	5	6	7	8	9	10
s_i	13	9	29	3	26	13	1	17	10	1	13
f_i	15	24	30	7	28	20	6	27	29	13	15

Una solución válida: {3,8,2}

Una solución óptima: {9,5,4,2}

- Función de selección óptima:
  - selección en orden creciente de terminación: {6,0,7,2}

3 Sistema GreedEx

- Programa:

```
public static boolean[ ] seleccionActividades (int[ ] c, int[ ] f) {
    boolean[ ] s = new boolean [c.length];
    s[0] = true;
    int i = 0;
    for (int j = 1 ; j < c.length ; j++){
        if (c[j]>=f[i]){
            s[j] = true;
            i = j;
        }
        else
            s[j] = false;
    }
    return s;
}
```

- Orden de complejidad: $O(n)$

  Considerando la ordenación inicial: $O(n \log n)$

3 Sistema GreedEx

- No encontramos ninguna visualización común, no trivial y útil de los algoritmos voraces
- Encontramos que lo más interesante era proponer funciones de selección

3 Sistema GreedEx

- Experimentación:
  - Descubrir cuáles son las funciones de selección óptimas para cierto problema:
    - partimos de un algoritmo voraz genérico
    - proponemos varias funciones de selección y aplicamos el algoritmo particularizado para ellas, y
    - evaluamos la optimidad de (comparamos) estas funciones de selección

3 Sistema GreedEx

- Aplicar las distintas funciones de selección a varios conjuntos de datos:
  - Acumulación de evidencia
  - Contrajemplos

3 Sistema GreedEx

- http://www.lite.etsii.urjc.es/tools/greedex/

3 Evaluaciones de GreedEx

- Asignatura:
  - "Diseño y análisis de algoritmos", 3º Ingen. Informática
- Práctica:
  - 1ª sesión (experimentación)
  - 2ª sesión (revisión)
- Entrega de informe tras sesión

Funciones de selección[1] óptimas
Proponga las funciones de selección (si existe alguna) que considere óptimas para resolver el problema de la selección de actividades.

Función de selección 1: XXXXX
Función de selección 2: XXXXX
…

Justificación de la función de selección XXXXX (repítase por cada función de selección propuesta)

Justificación razonada: Se explica en términos coloquiales porqué la función de selección es óptima. Obsérvese que se espera un razonamiento sobre el resultado (óptimo) de aplicar la función de selección, no una explicación de cómo funciona o se ejecuta.

Evidencia experimental obtenida con GreedEx (resumen): Resumen de los datos de entrada probados con GreedEx y que proporcionan evidencia experimental de la optimidad de la función de selección:

Tabla de resumen o tabla abreviada:

Evidencia experimental obtenida con GreedEx (ejemplos detallados): Se incluye la ejecución detallada del algoritmo con diversas funciones de selección y varios datos de entrada, de forma que sean ilustrativos del comportamiento óptimo de las propuestas y del comportamiento no óptimo de las descartadas.

Tabla de datos de entrada:

Tabla de resultados con todas las funciones de selección ejecutadas:

Ejecución de cada función de selección: (opcional, repítase por cada una)

3 Evaluaciones de GreedEx

- Recogida de datos de los informes:
  - Transcripción de datos “relevantes”
- Análisis de los informes:
  - Análisis cualitativo sin categorías previas (“grounded theory”)
  - Proceso intensivo e iterativo, iterativo…
    - Repetidas lecturas de cada informe
    - 3 rondas de análisis

3 Evaluaciones de GreedEx

- Factores distintivos de las respuestas incorrectas:
  1. Propuesta de funciones de selección subóptimas
  2. Incoherencia del razonamiento
  3. Criterio de optimización adicional:
     - Maximizar ocupación de la sala
     - Minimizar tiempo de espera
  4. Propuesta dependiente de los datos de entrada
- Algunos malentendidos graves

3 Evaluaciones de GreedEx

- Categorías de respuestas:

Categoría	Factores
A	–
B	1
C	1-2
D	1-3
E	1-2-3
F	1-3-4
G	1-2-3-4

- Grado variable de viabilidad de sus modelos mentales

3 Evaluaciones de GreedEx

- Intervenciones didácticas para reducir malentendidos:
  - Ligera modificación del enunciado y la plantilla de informe
  - Elaboración de apuntes
  - Adaptación de las clases:
    - Énfasis en propuesta de funciones de selección
  - Uso del método didáctico con problemas no resolubles con algoritmos voraces
  - Inclusión de una sesión de entrenamiento
- Dos evaluaciones en cursos siguientes

3 Evaluaciones de GreedEx

- Evolución de las categorías, como consecuencia de las intervenciones:

Categorías	1ª eval.	2º eval.	3º eval.
A	22.22%	28.57%	47.83%
B	5.56%	23.81%	39.13%
C	16.67%	23.81%	13.04%
D	11.11%	19.05%	0%
E	16.67%	4.76%	0%
F	22.22%	0%	0%
G	5.56%	0%	0%

- Las categorías menos viables desaparecen:

	1st eval.	2nd eval.	3rd eval.
A-B	27.78%	52.38%	86.96%
C-D	27.78%	52.86%	13.04%
E-F-G	44.45%	4.76%	0%

4 Sistema OptimEx

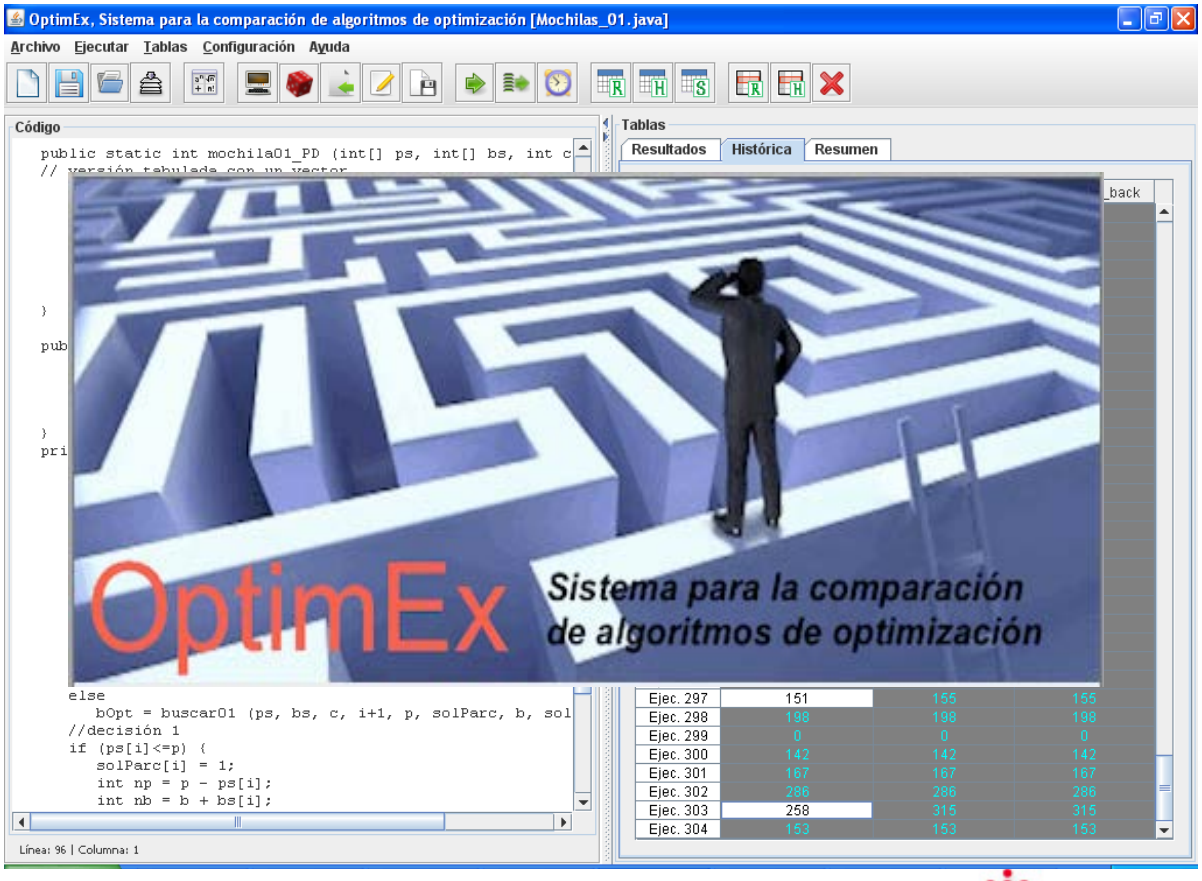
- Sistema general de comparación de la optimidad de algoritmos que resuelven el mismo problema
- Secuencia de uso:
  1. GreedEx: introducción a la experimentación sobre optimización con algoritmos voraces seleccionados
  2. OptimEx: experimentación con algoritmos de optimización cualesquiera

4 Sistema OptimEx

- Diseño:
  - Interfaz de usuario similar a GreedEx
  - Se eliminaron elementos dependientes del problema:
    - Panel de visualización y control de animación
    - Panel de teoría
  - Para soportar cualquier algoritmo de optimización:
    - Compilar y ejecutar

4 Sistema OptimEx

- http://www.lite.etsii.urjc.es/tools/greedex/



4 Evaluaciones de OptimEx

- Asignatura:
  - "Algoritmos avanzados", 4º G. Ingeniería Informática, 2 grupos
  - Prácticas:

1. Voraz	Problema de selección de actividades
2. Vuelta atrás, ramificación y poda	Íd. ponderadas
4. Programación dinámica	Íd. ponderadas
5. Algoritmos aproximados	Íd. ponderadas

    - Realizadas satisfactoriamente, además con posibilidad de mejorarlas

4 Evaluaciones de OptimEx

- Práctica 5:
  - Doble objetivo, específico y englobador:
    - Desarrollo de un algoritmo heurístico (voraz)
    - Comparación de la optimalidad de varios algoritmos para el mismo problema
  - Resultados esperados:
    - Algoritmos heurísticos: subóptimos
    - Otros: óptimos

4 Evaluaciones de OptimEx

- Análisis cualitativo:
  - Contenido
  - Lenguaje
  - Comentarios abiertos
- Fases de análisis:
  1. Ronda exploratoria
  2. Elaboración de tabla y nueva ronda
  3. Aclaración de preguntas y nueva ronda

4 Evaluaciones de OptimEx

- Dificultades:

Resultado	Número de grupos	% de grupos
Correcto	8	28'6%
Parcial	2	7'1%
Incorrecto	18	64'3%

- Errores:
  - 13 grupos: ningún algoritmo dio resultados óptimos
  - 4 grupos: marcaron un algoritmo subóptimo como óptimo
  - 1 grupo: no identificó ningún algoritmo como óptimo

4 Evaluaciones de OptimEx

- Resumen de hallazgos:
  - Mayoría de grupos con la experimentación mal
  - Frecuente uso de materiales inadecuados y mala interpretación de las tablas
  - Frecuente impercepción de resultados incorrectos ni de dificultades
  - Satisfacción por organización de prácticas
  - Frecuente uso incorrecto del término “óptimo”

4 Evaluaciones de OptimEx

- Segunda evaluación:
  - Modificaciones menores del enunciado
  - Explicación en la guía de usuario de errores posibles
  - Se permitió que corrigieran la práctica
- Resultados:
  - Mejora de práctica correcta (del 35’7% al 51’7%)
  - Otros problemas similares a la evaluación anterior

4 Evaluaciones de OptimEx

- Tercera evaluación:
  - Distribución de la comparación en todo el curso
  - Prácticas:

2. Algoritmos aproximados	Problema de plan óptimo de sedes
3. Vuelta atrás, ramificación y poda	Ídem
5. Programación dinámica	Ídem

    - Comparación de resultados en prácticas 2, 3b y 5b
    - En todas las prácticas podían hacer una segunda entrega

4 Evaluaciones de OptimEx

- Resultados:
  - Alto porcentaje de prácticas correctas de la práctica 2 (75%; tras segunda entrega, 95%):
    - La mayor parte de las prácticas incorrectas, por no entender bien la descripción de los algoritmos propuestos
  - Un tercio propone algoritmos mejores
  - Frecuente interés por análisis más detallado por casos o de la eficiencia

4 Evaluaciones de OptimEx

- Resultados:
  - Disminución progresiva de entregas en las prácticas 3b y 5b
  - Porcentajes mayores de resultados correctos para las prácticas 3b y 5b (88% y 89%)
  - Algunos alumnos realizan la experimentación sin pedírsela (prácticas 3a y 5a)

5 Conclusiones

- Sistemas de visualización o experimentación:
  - Su uso permite realizar tareas que de otra forma serían muy laboriosas o imposibles

5 Conclusiones

- Uso de sistemas en educación:
  - Aumentan la motivación
  - Quizá oportunidad para identificar dificultades de los alumnos:
    - Conceptos aparentemente sencillos quizá no lo son (p.ej. de optimización)
  - ¿Porqué no hemos encontrado dificultades conceptuales con SRec?
    - Evitar respuesta directa y trivial

5 Conclusiones

- Uso de sistemas en educación:
  - Integración de herramientas informáticas, contenidos de las clases, actividades docentes y materiales docentes:
    - Revisión y ajuste continuo
  - Mejora de la práctica docente:
    - Convenientes sesiones de laboratorio de introducción y de corrección de fallos
    - Es preferible una experimentación “dirigida” con varias prácticas de dificultad pequeña que una sola experimentación “abierta”

5 Conclusiones

- Otros comentarios:
  - Resultados explicables por la teoría pero imprevisibles en principio
  - Cualquier profesional puede hacer investigación:
    - Siempre hay trabajos futuros…

Bibliografía

- T. L. Naps, G. Roessling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger y J. Á. Velázquez-Iturbide, “Exploring the role of visualization and engagement in computer science education”, *ACM SIGCSE Bulletin*, 35(2):131-152, junio 2003, DOI 10.1145/960568.782998
- Velázquez-Iturbide, J. Á., Pérez-Carrasco, A., Urquiza-Fuentes, J. “SRec: An animation system of recursion for algorithm courses”, *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE 2008, pp. 225-229, DOI 10.1145/1384271.1384332
- Velázquez-Iturbide, J. Á., Pérez-Carrasco, A. “How to use the SRec visualization system in programming and algorithm courses”, *Inroads*, en imprenta

Bibliografía

- Velázquez-Iturbide, J. Á., Debdi, O., Esteban-Sánchez, N., Pizarro, C., "GreedEx: A visualization tool for experimentation and discovery learning of greedy algorithms", *IEEE Transactions on Learning Technologies*, 6(2):130-143, abril-junio 2013, DOI 10.1109/TLT.2013.8
- Velázquez-Iturbide, J. Á., "An experimental method for the active learning of greedy algorithms", *ACM Transactions on Computing Education,* 13(4), artículo 18, 23 págs., octubre 2013, DOI 10.1145/2534972
- Velázquez-Iturbide, J. Á., "GreedEx and OptimEx: Two tools to experiment with optimization algorithms", *International Journal of Engineering Education*, 32(3A):1.097-1.106, 2016

¡Muchas gracias!