

Grados en Ingeniería Informática e Ingeniería de Computadores
Asignatura Algoritmos Avanzados
Curso 2022/2023
Práctica nº 2

Objetivo

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

Carácter

La realización de la práctica es voluntaria, pero su entrega es un requisito para poder entregar parte de las prácticas 3 y 5. Puede hacerse individualmente o en pareja.

Enunciado

Se dispone de un computador de alto rendimiento con capacidad para procesar varios exabytes de datos al día. Se conocen con antelación las cantidades de datos que deberían procesarse en n días, uno a uno, de forma que el día i deben procesarse x_i exabytes.

Por cada exabyte procesado se cobra una cantidad fija, pero si se dejan datos sin procesar, ni se cobra por ellos ni pueden procesarse al día siguiente (“se pierden”). No siempre pueden procesarse los exabytes solicitados debido a las restricciones del sistema, que sólo puede procesar una determinada cantidad de exabytes al día. De hecho, el sistema presenta el problema de degradación de su rendimiento, de forma que cada día puede procesar menos cantidad de datos que el día anterior. El primer día pueden procesarse c_1 exabytes, el segundo día c_2 , y así sucesivamente hasta c_n , de forma que $c_1 > c_2 > \dots > c_n$. Puede restaurarse la capacidad potencial de procesamiento del sistema si se reinicia, pero ese día ya no pueden procesarse datos algunos.

Sean las cantidades de datos d_1, d_2, \dots, d_n a procesar en los próximos n días y el perfil del rendimiento del sistema $c_1 > c_2 > \dots > c_n$ en n días a partir de su puesta en marcha. El *problema de la planificación maximal de procesamiento de datos* consiste en determinar en qué días se reinicializará el sistema, de forma que se maximice la cantidad de datos procesada en los n días. Se supone que el sistema está recién arrancado el primer día.

Por ejemplo, sean los siguientes valores para 4 días:

	día 1	día 2	día 3	día 4
d	10	1	7	7
c	8	4	2	1

La mejor solución consiste en reiniciar el sistema el segundo día. De esta forma, puede procesarse un total de 19 exabytes:

- día 1: 8 exabytes (la capacidad máxima del primer día),
- día 2: 0 exabytes (para su reinicio),
- día 3: 7 exabytes (ya que tras su reinicio podría procesar hasta 8), y
- día 4: 4 exabytes (la capacidad máxima del segundo día tras un reinicio).

Obsérvese que sin realizar ningún reinicio, se procesarían $8+1+2+1=12$ exabytes. Asimismo, cualquier otra planificación produce una solución submaximal.

Pueden concebirse varios algoritmos heurísticos. Sea un algoritmo H1 basado en la asunción de que debe reiniciarse el sistema cuando no puede procesar todos los datos asignados para ese día, ya que es una señal de que el rendimiento del sistema se ha degradado.

En el ejemplo anterior, H1 reiniciaría el sistema el primer día (ya que $8 < 10$), después procesaría 1 exabyte el segundo día, volvería a reiniciar el sistema el tercer día (ya que $4 < 7$) y finalmente procesaría los 7 exabytes del cuarto día. En total, se procesarían $0+1+0+7=8$ exabytes.

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

El problema puede resolverse con la siguiente cabecera:

```
public static int procesar (int[] ds, int[] cs)
```

donde *ds* es un vector con los datos a procesar en los *n* días y *cs* un vector con la capacidad de procesamiento del sistema en *n* días desde su reinicio. Recuerdese que $c_1 > c_2 > \dots > c_n$. Para el ejemplo anterior, la llamada será *procesar*($\{10,1,7,7\}, \{8,4,2,1\}$). El método debe devolver la cantidad total de datos procesados para la planificación que realice el algoritmo. Para el ejemplo, una planificación maximal devuelve 19.

Informe

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

1. **Especificación del problema.** Debe desarrollarse una especificación del problema, identificando sus distintas partes: precondition y poscondition (formada a su vez por condición de validez y función objetivo). No hace falta utilizar notación matemática, pero la respuesta debe ser precisa y completa.
2. **Diseño e implementación de algoritmos heurísticos.** Se debe diseñar al menos un algoritmo heurístico adicional al esbozado en el enunciado de la práctica. Los nuevos algoritmos deben ser “razonables”, es decir, deben basarse en alguna justificación intuitiva de por qué pueden conducir a soluciones optimales, como la presentada en el primer párrafo de esta página. Posteriormente, deben implementarse el algoritmo del enunciado y los diseñados por el alumno en una sola clase de Java, todos con la cabecera anterior (obviamente, con distintos identificadores de método).
3. **Experimentación con la optimalidad de los algoritmos.** Se comparará con AlgorEx la optimalidad de los algoritmos desarrollados. Se aconseja leer, al menos, el uso de AlgorEx descrito en los apuntes sobre experimentos algorítmicos, disponibles en el aula virtual. Debe aportarse la siguiente información:
 - Material del experimento. Se debe:
 - Identificar cada algoritmo heurístico.
 - Indicar los rangos de valores y características usados para la generación aleatoria de los datos de entrada.
 - Conclusión. Razonar si cada algoritmo es exacto o no según los resultados de la experimentación.

- Evidencias. Deben aportarse los resultados recogidos en la tabla de resumen numérico y en los diagramas de resumen gráfico, explicando su significado.
 - Contraejemplos. Para cada algoritmo inexacto, se deben aportar dos contraejemplos, uno encontrado en el experimento y otro más sencillo deducido tras analizar las características de los contraejemplos encontrados experimentalmente. Explica las características de los contraejemplos que producen que el algoritmo sea subóptimo.
 - Incidencias (opcional) que hayan dificultado la realización del experimento.
4. **Conclusiones.** Se presentan las conclusiones obtenidas tras realizar la práctica. Estas conclusiones pueden consistir en una valoración de los algoritmos heurísticos, la experiencia de uso de AlgorEx o cualquier comentario sobre la práctica. Por ejemplo, pueden describirse incidencias que han dificultado su realización, sus aspectos más atractivos o más difíciles, sugerencias, etc.

Entrega

El alumno debe entregar el informe por medio del apartado de Evaluación del aula virtual. Si se tienen dificultades, puede enviarse por el correo del aula virtual con el asunto “Práctica 2”. El plazo de entrega es el domingo 16 de octubre de 2022, incluido.

Evaluación

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