

Grados en Ingeniería Informática e Ingeniería de Computadores

Asignatura Algoritmos Avanzados

Curso 2021/2022

Práctica nº 2

Objetivo

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

Carácter

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

Enunciado

Un estudiante dispone de d días para preparar los exámenes finales de a asignaturas, $a \leq d < 2 \cdot a$. La calificación en cada asignatura siempre es mayor que cero y menor o igual a diez. Para que su rendimiento sea óptimo, sabe que cada día debe dedicarlo exclusivamente a una asignatura, debiendo dedicarle al menos un día. El *problema de la asignación maximal de calificaciones* consiste en desarrollar un algoritmo que asigne el tiempo que debe dedicar a cada asignatura de forma que la suma de todas las calificaciones (es decir, la calificación global) sea máxima.

Por ejemplo, con seis días y cuatro asignaturas, el estudiante debe dedicar al menos un día a cada asignatura, quedándole dos días de margen. Puede dedicar estos dos días a una sola asignatura o dedicar un día extra a dos asignaturas. Así, dada la siguiente asignación de puntuaciones:

#días \ asignatura	A1	A2	A3	A4
1	4	3	5	2
2	4	5	6	4
3	5	6	8	7

una solución optimal es 19, resultante de obtener las calificaciones 4+3+5+7 (un día de estudio para las tres primeras asignaturas y 3 para la cuarta). Puede comprobarse que cualquier otra asignación producirá una calificación menor.

En el ejemplo se ve que a mayor número de días de estudio por asignatura, mayor es la puntuación prevista. Sin embargo, para facilitar la generación de datos de entrada aleatorios, podemos suponer que no existe esta tendencia.

Pueden concebirse varios algoritmos heurísticos. Un primer algoritmo H1 seleccionará secuencialmente la mejor opción posible para cada asignatura, mientras haya “días extra” de estudio. Otro algoritmo H2 podría realizar una planificación equitativa de los días extra entre todas las asignaturas.

En el ejemplo anterior, H1 asignaría tres días a A1, debiendo ya dedicarse solamente un día a cada una de las demás asignaturas. La puntuación global sería $5+3+5+2=15$. Asimismo, con H2, podrían dedicarse dos días a las dos primeras asignaturas, resultando una calificación global de $4+5+5+2=16$.

El objetivo de la práctica es implementar, al menos, ambos 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 asigs (int[][] cs)
```

donde *cs* es una matriz de calificaciones, tal que *cs*[*d*][*a*] contiene la calificación obtenida en la asignatura *a* si se le dedican *d* días. Para el ejemplo anterior, la llamada será *asigs*({{4,3,5,2},{4,5,6,4},{5,6,8,7}}). El método debe devolver la calificación global correspondiente a la planificación realizada. Para el ejemplo anterior, la calificación optimal es 19.

Recuérdese la restricción $a \leq d < 2 \cdot a$. En el ejemplo anterior, $a=4$, $d=6$ ($4 \leq 6 < 8 = 2 \cdot 4$), de manera que el número de días a asignar a cada asignatura puede oscilar entre 1 y 3. Dado que debe dedicarse al menos un día a cada asignatura, el número *e* de días extra de estudio debe cumplir $0 \leq e < a$, es decir $0 \leq e \leq a-1$. Esta restricción se cumple si la matriz *cs* tiene un número *f* de filas tal que $1 \leq f \leq a$. A la inversa, los números de días totales y de días extra pueden deducirse de las dimensiones de la matriz.

Informe

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

1. **Especificación del problema.** Debe desarrollarse una especificación del problema, identificando sus distintas partes: precondition y poscondition (formada a su vez por condición de validez y función objetivo). No hace falta utilizar notación matemática, pero la respuesta debe ser precisa y completa.
2. **Implementación de los algoritmos.** Implementar ambos algoritmos en una sola clase de Java y con la cabecera anterior (obviamente, con distintos identificadores de método).

Se pueden presentar algoritmos heurísticos adicionales que sean “razonables”. Para cada nuevo algoritmo, se explicará claramente el criterio voraz usado y una justificación intuitiva de por qué dicho criterio puede conducir a soluciones optimales. La cabecera de estos nuevos algoritmos debe ser igual a la anterior (aunque, obviamente, con distintos identificadores de método).

3. **Experimentación con la optimalidad de los algoritmos.** Se comparará con AlgorEx la optimalidad de los algoritmos desarrollados. Se aconseja leer, al menos, la guía de usuario resumida disponible en el aula virtual. Debe aportarse la siguiente información:
 - Material del experimento. Se debe:
 - Identificar cada algoritmo heurístico.
 - Indicar los rangos de valores usados para la generación aleatoria de los datos de entrada. (Opcionalmente, el experimento puede repetirse para diversos números de filas y de columnas y así comprobar si cada heurística mejora o empeora con estas variaciones).

- 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 17 de octubre de 2021, incluido.

Evaluación

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