

Práctica I

Algoritmos Voraces

Metodología y Tecnología de la Programación

Curso 2004/05

Grupos A, B y C

1. Enunciado

Un operario de la construcción tiene una serie de pequeños trabajos a realizar numerados desde 1 hasta N . Dispone de un determinado tiempo T de trabajo. Cada trabajo i tiene un beneficio b_i y se realiza en un tiempo d_i , pero hay ciertos trabajos que no se pueden realizar inmediatamente después de otros, esta restricción viene determinada por una matriz de dimensiones $N \times N$ compuesta de valores booleanos, donde $M[i, j] = true$ indica que el trabajo i no se puede realizar inmediatamente después del trabajo j y $M[i, j] = false$ indica que el trabajo i sí se puede realizar inmediatamente después del trabajo j .

¿Cómo debe realizar el operario los trabajos encargados de forma que maximice el beneficio obtenido?

Observación: los trabajos se pueden realizar parcialmente.

1.1. Ejemplo ilustrativo

Considerar el siguiente ejemplo ilustrativo del problema planteado con los siguientes valores:

$N=5$, $T=8$, $B = (5,4,6,5,4)$, $D = (2,3,2,1,1)$

false	false	false	false	false
false	false	false	false	true
false	false	false	false	false
false	false	false	false	false
false	false	true	true	false

Cuadro 1: Matriz M

El trabajo que está en la quinta posición no puede realizarse inmediatamente después del trabajo que está en la tercera posición.

El trabajo que está en la quinta posición no puede realizarse inmediatamente después del trabajo que está en la cuarta posición.

El trabajo que está en la segunda posición no puede realizarse inmediatamente después del trabajo que está en la quinta posición.

Con los criterios de entrada anteriormente descritos se obtendría la siguiente solución.

Secuencia de trabajos que optimizan el beneficio:

Trabajo de beneficio 5 y duración 1

Trabajo de beneficio 6 y duración 2

Trabajo de beneficio 5 y duración 2

Trabajo de beneficio 4 y duración 1

Beneficio: 20.

2. Se pide

1. Hacer un algoritmo voraz que resuelva este problema maximizando el beneficio.
2. Definir claramente los elementos esenciales del esquemas
3. Estudiar la complejidad del algoritmo del apartado anterior
4. Implementar dicho algoritmo en un lenguaje de programación, por ejemplo Pascal, C++ o Java.
5. Estudiar el tiempo de ejecución de la implementación correspondiente al apartado 4) comparándolo con el apartado 3). Para realizar dicha comparación se dibujará una gráfica que muestre las curvas temporales de ambas.

3. Entrega

1. La fecha tope para entregar la práctica será hasta las 20:00 horas del día 13 de Diciembre.
2. La práctica se entregará a través del campus virtual en un archivo zip etiquetado como Apel1Apel2Prac1.zip. Así si el alumno se llama Juan Pepito Perico, entonces el archivo sería PepitoPericoPrac1.zip. El archivo debe contener:
 - La resolución teórica del problema en un documento Word, pdf o ps.
 - Los fuentes y los ejecutables de la implementación de la práctica.
 - Gráficos.

3. La evaluación de la práctica consistirá en la defensa de la misma por parte del alumno ante el profesor, en dicha defensa pueden estar presentes otros profesores de la asignatura.
4. El día y hora de la defensa de cada alumno será publicada en una lista el día 15 de Diciembre.
5. La defensa de la práctica será realizada mediante una presentación usando transparencias.
6. La máxima puntuación que se puede obtener con este práctica será proporcional al cociente entre el número de pruebas que se realicen en este cuatrimestre y el valor máximo asociado a este tipo de prácticas, que es de dos puntos.