PROGRAMACIÓN Y ESTRUCTURAS DE DATOS AVANZADAS

Febrero 2018 (Primera semana)

Normas de valoración del examen:

- La nota del examen representa el 80% de la valoración final de la asignatura (el 20% restante corresponde a las prácticas).
- Cada cuestión contestada correctamente vale 1 punto.
- Cada cuestión contestada incorrectamente baja la nota en 0.3 puntos.
- Debe obtenerse un mínimo de 3 puntos en las cuestiones para que el problema sea valorado (con 3 cuestiones correctas y alguna incorrecta el examen está suspenso).
- La nota total del examen debe ser al menos de 4.5 para aprobar.
- Las cuestiones se responden en una hoja de lectura óptica.

SOLUCIONES:

Test:

Tipo A: 1A 2C 3B 4C 5D 6C
Tipo B: 1B 2C 3A 4C 5D 6C

Problema (4 puntos).

Tenemos *n* objetos de volúmenes v_1... v_n, y un número ilimitado de recipientes iguales con capacidad *R* (con $v_i \leq R$, para todo *i*). Los objetos se deben meter en los recipientes sin partirlos, y sin superar su capacidad máxima. Se busca el mínimo número de recipientes necesarios para colocar todos los objetos.

La resolución de este problema debe incluir, por este orden:

1. Elección razonada del esquema <u>más apropiado</u> de entre los siguientes: Voraz, Divide y Vencerás, Vuelta atrás o Ramificación y Poda.
   Escriba la estructura general de dicho esquema e indique como se aplica al problema (0,5 puntos).
2. Descripción de las estructuras de datos necesarias (0,5 puntos solo si el punto 1 es correcto).
3. Algoritmo completo a partir del refinamiento del esquema general (2.5 puntos sólo si el punto 1 es correcto). Si se trata del esquema voraz debe hacerse la demostración de optimalidad.
4. Estudio del coste del algoritmo desarrollado (0,5 puntos solo si el punto 1 es correcto).

Solución:

1. **El esquema más apropiado es ramificación y poda. El esquema general se encuentra formulado en el libro de texto de la asignatura, en la página 187.**

2. **Podemos representar el reparto de objetos entre recipientes mediante un vector en el que cada posición indique a qué recipiente se ha asignado cada objeto: *objetos = vector[1..n] of entero***

   La solución es la cantidad entera S de recipientes empleados.

   Utilizamos un montículo de mínimos en el que cada componente almacene una solución parcial (nodo) con su cota correspondiente:

   nodo = tupla
   asignaciones: vector[1..N] de enteros
   etapa: cardinal // objeto por el que se va procesando
   num-recip: cardinal // num de recipientes
   capacidad[1..N] de real // capacidades disponibles de cada envase utilizado

3. **La solución completa al problema se encuentra desarrollada en:**

   Estructuras de datos y métodos algorítmicos
   N. Martí Oliet, Y Ortega Mallén y J.A. Verdejo López
   Prentice Hall
   Ejercicio 15.9

   Para el desarrollo se utilizan las siguientes estimaciones

   - **cota inferior: el número de envases ya utilizados en la solución parcial**

   - **cota superior: Se hace una estimación metiendo cada objeto de los pendientes en el primer recipiente en el que quepa. Si no cabe en ninguno, se añade un nuevo envase a la solución parcial.**

4. **El número de recipientes está limitado a n, es decir al número de objetos. Una estimación del coste es el tamaño del árbol, que en el peor caso crece como O(n!), ya que cada nodo del nivel k puede expandirse con los n-k objetos que quedan por asignar a recipientes.**