examenfinal2015-06-aunque-diga-e...

Anónimo

Algoritmos y Estructuras de Datos II

2º Grado en Ingeniería Informática

Facultad de Informática
Universidad de Murcia

No entregar esta hoja con el examen.
La primera pregunta cuenta un 30% de la nota de teoría.
El resto de preguntas valen el 70% (1,4 puntos cada una).

1. Considerar el siguiente procedimiento recursivo:

```
operación crowdfund(people: array [1..n] de entero; n: entero;);
 index:= n;
 mientras (index>=2) Y (people[index]>1) hacer
    people[index] = people[index] - 1;
    index:= index/2:
 finmientras
 si n<4 entonces return 1
 si no
   amount := 0;
   amount := amount + crowdfund(n/4);
   index := 1;
   mientras (index <= 2)
     amount:= amount + crowdfund(n/2,people);
     index := index + 1;
   finmientras
   return  amount
finsi
```

Estimar el tiempo de ejecución del algoritmo en el mejor y en el peor caso. Expresar el tiempo usando las notaciones asintóticas adecuadas. A partir del resultado indicar si existe un orden exacto para el tiempo del algoritmo en todos los casos. Indicar cuántos y qué casos base se podrían utilizar para resolver las constantes.

2. Divide y vencerás

Una caja con n bombones se considera aburrida si se repite un mismo tipo de bombón (denominado bombón pesado) más de n/2 veces. Escribir un algoritmo basado en el esquema Divide y vencerás que decida si una caja es o no aburrida y devuelva (en su caso) el tipo de bombón que le confiere dicha propiedad. El coste debe ser a lo sumo O(n log n). NOTA: si una caja tiene un bombón pesado, entonces necesariamente también lo es al menos una de sus mitades.

3. Programación dinámica

La **Liga Estelar** es una competición en la que los participantes disponen de un presupuesto *P* para configurar un equipo de fútbol sala a partir de un listado dado de jugadores. De cada jugador se conoce la posición en la que puede jugar, que puede ser portero, defensa, centrocampista o delantero. El equipo debe tener exactamente cinco jugadores, respetando que debe haber un portero (solo uno) y cuatro jugadores del resto las posiciones restantes. Para cada jugador *j*, se conoce el precio *pj* de su ficha, y su valoración *vj* ($vj \in [0,10]$) obtenida según sus resultados en las ligas anteriores.

Diseñar un algoritmo basado en Programación dinámica que determine los cinco jugadores que deben ficharse para conformar el equipo, sin sobrepasar un presupuesto dado *P* y maximizando la valoración total del equipo, es decir, la suma de las valoraciones de los jugadores seleccionados. Describir en detalle la función de recurrencia, los casos base, las tablas, la forma de rellenarlas y la forma de reconstruir la solución.

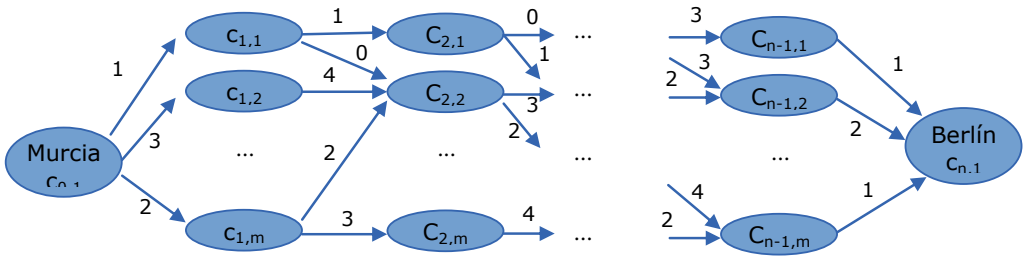
ENUNCIADO DE PROBLEMA para las preguntas 3-6:

Blablagraph:

Tenemos que ir de Murcia a Berlín en coche. Haremos el viaje en n etapas, que numeramos de 1 a n. Al final de cada etapa i hay m ciudades, $c_{i,j}$, i=1..n y j=1..m, excepto en la etapa n, con una sola ciudad, Berlín, $c_{n,1}$. El origen, Murcia, es $c_{0,1}$. Desde cualquier c_{ij}, excepto Berlín, se puede llegar al menos a una ciudad de la siguiente etapa, pero en general no a cualquier ciudad.

Como la crisis obliga, nos ofreceremos a llevar pasajeros por un módico precio en la plataforma online *blablagraph*. Asumimos que todos los trayectos de todas las etapas tienen la misma longitud. Según llevemos 0, 1, 2, 3 ó 4 pasajeros cada trayecto nos costará 30, 20, 10, 0 ó -10 euros, respectivamente (3 pasajeros ya nos pagan el combustible y con 4 hasta sacamos dinero).

Tenemos un grafo multietapa, G, que nos indica cuántos pasajeros vendrían en cada trayecto entre dos ciudades en etapas consecutivas:



4. Diseñar un algoritmo voraz que encuentre una buena forma de resolver el problema. Hay que ajustarse al esquema y desarrollar sus funciones. ¿Garantiza ese algoritmo la solución óptima?

5. Resolver el problema de forma óptima por backtracking. Se deberán utilizar los esquemas vistos en clase. Definir la forma de representar la solución, el tipo de árbol usado, el esquema y las funciones genéricas del esquema. Seguir los pasos de desarrollo vistos en clase.

6. Resolver el problema de forma óptima por ramificación y poda. Se deberán utilizar los esquemas vistos en clase. Definir la forma de representar la solución, la forma de calcular las cotas y la estimación de beneficio, así como las estrategias de ramificación y poda. Seguir los pasos de desarrollo vistos en clase.