Algoritmos y Estructuras de Datos II - 24 de junio de 2013
Segundo Parcial - Recuperatorio

Alumno: .....

Recuerde utilizar una hoja aparte para la solución de cada ejercicio. La cátedra le proveerá las hojas que solicite. Recuerde también explicar y justificar con claridad cada una de sus respuestas.

1. El siguiente es el algoritmo de Dijkstra, que para cada vértice w devuelve en D[w] el costo del camino de menor costo que va desde el vértice v al vértice w, asumiendo que los costos de las aristas son números naturales y están dados por la matriz L:

```
fun Dijkstra(L: array[1..n,1..n] of nat, v: {1,...,n}) ret D: array[1..n] of nat
    var c: {1,...,n}
    C:= {1,2,...,n}-{v}
    for j:= 1 to n do D[j]:= L[v,j] od
    do n-2 times → c:= elemento de C que minimice D[c]
                   C:= C-{c}
                   for j in C do D[j]:= mín(D[j],D[c]+L[c,j]) od
    od
end fun
```

Explique y justifique claramente lo que calcula el siguiente programa, que como puede observar es una variante del anterior:

```
fun ¿qué_hago?(L: array[1..n,1..n] of nat, v: {1,...,n}) ret D: array[1..n] of nat
    var c: {1,...,n}
    C:= {1,2,...,n}
    for j:= 1 to n do D[j]:= ∞ od
    D[v]:= 0
    do n-1 times → c:= elemento de C que minimice D[c]
                   C:= C-{c}
                   for j in C do D[j]:= mín(D[j],D[c]+L[c,j]) od
    od
end fun
```

Explique y justifique claramente también lo que calcula el siguiente programa, que recibe un conjunto X de vértices en vez de uno solo:

```
fun ¿y_qué_hago_yo?(L: array[1..n,1..n] of nat, X ⊆ {1,...,n}) ret D: array[1..n] of nat
    var c: {1,...,n}
    C:= {1,2,...,n}
    for j:= 1 to n do D[j]:= ∞ od
    for j ∈ X do D[j]:= 0 od
    do n-1 times → c:= elemento de C que minimice D[c]
                   C:= C-{c}
                   for j in C do D[j]:= mín(D[j],D[c]+L[c,j]) od
    od
end fun
```

2. Considere la siguiente variante del problema de la mochila, en la que se cuenta con dos mochilas en vez de una: sean n objetos de peso $w_1, \ldots, w_n$ y valor $v_1, \ldots, v_n$ y dos mochilas de capacidad W y Z. Se desea hallar el máximo valor total obtenible con la carga de las dos mochilas sin exceder sus capacidades.

   a) Comprobar que no es posible resolver óptimamente este problema cargando primero una de las mochilas (con cualquiera de los algoritmos provistos para el problema de la mochila habitual sin fraccionamiento) y luego la otra. Puede valerse del siguiente ejemplo: $W = 8$, $Z = 8$, $n = 5$, $v_1 = 5$, $v_2 = 5$, $v_3 = 4$, $v_4 = 4$, $v_5 = 6$, $w_1 = 3$, $w_2 = 3$, $w_3 = 5$, $w_4 = 5$, $w_5 = 6$.

   b) Se define en general $m(i, j, k)$ = "máximo valor alcanzable con los objetos $1, \ldots, i$ sin exceder la capacidad j de la primera mochila, ni la capacidad k de la segunda". Dar una definición recursiva (backtracking) de $m(i, j, k)$ para $0 \leq i \leq n$, $0 \leq j \leq W$ y $0 \leq k \leq Z$.

   c) Con la definición así obtenida, calcular $m(n, W, Z)$ en el ejemplo del primer inciso.

3. Para un cierto problema se ha hallado la siguiente definición utilizando backtracking:

$$m(i,j) = \begin{cases} 5 & j = 0 \\ 2 & j > 0 \wedge i = 0 \\ j + m(i-1, j-1) & i > 0 \wedge j > 0 \wedge i = n \\ \text{mín}(1 + m(i, j-1), m(i, j-i), m(i+1, j-1)) & i > 0 \wedge j \geq i \wedge i \leq n-1 \\ \text{mín}(1 + m(i-1, j), m(i, 0), m(i, j-1)) & c.c. \end{cases}$$

Además la llamada principal es $m(n, k)$.

Se pide transformar esta definición en una iterativa utilizando una tabla de valores calculados, es decir, usando programación dinámica. Recuerde asegurarse de que las celdas de la tabla deben calcularse en un orden tal que aquéllas celdas que participan del cálculo de otra deben calcularse previamente.

4. Un árbol binario es perfectamente balanceado si todos sus caminos maximales desde la raíz tienen la misma longitud. Ejemplifique y explique:

   a) Dar un ejemplo de árbol binario de búsqueda perfectamente balanceado con 15 elementos (que sean los números del 1 al 15).

   b) ¿Cuántos heaps perfectamente balanceados hay con 3 elementos (números del 1 al 3)?

   c) ¿Cuántos heaps perfectamente balanceados hay con 7 elementos (números del 1 al 7)?