Algoritmos y Estructuras de Datos II - 21 de junio de 2012

Parcial 3: Algoritmos avanzados

Docentes: Daniel Fridlender y Silvia Pelozo

Alumno: .....

No se olvide de **justificar** con la mayor **claridad** posible **todas** sus soluciones.

1. Una refinería ha recibido solicitud de combustible por parte de n estaciones de servicio. Se cuenta con un único camión de distribución y cada estación ha solicitado exactamente una carga completa (un camión). Por ello, se debe regresar a la refinería después de aprovisionar cada estación. Lamentablemente el día no alcanza para atender todas las solicitudes, por lo que la refinería se propone atender el mayor número de solicitudes posibles en el día. Dé un algoritmo voraz que devuelva el conjunto de estaciones que deben ser atendidas teniendo en cuenta que:

   - el camión se encuentra inicialmente vacío y en la refinería,
   - el camión debe finalizar el día vacío y en la refinería,
   - el tiempo de carga del camión es c,
   - el tiempo de descarga del camión en la estación i es d_i,
   - el tiempo del viaje de ida de la refinería a la estación i es t_i y es igual al tiempo del viaje de regreso,
   - el tiempo total de que se dispone en el día es T, no puede excederse.

   Se puede asumir que los tiempos están dados en minutos.

2. El problema de la moneda consiste en encontrar, para un conjunto de denominaciones $d_1, \ldots, d_n$ el menor número de monedas necesarias para abonar exactamente un monto dado M. Se asume que las denominaciones y el monto son números naturales arbitrarios, por lo que se descarta la solución voraz, que sólo funciona para ciertos conjuntos de denominaciones.

   Se plantea una solución diferente a la presentada en clase. En vez de definir $m(i, j) =$ "menor número de monedas de denominacion $d_1, \ldots, d_i$ necesarias para pagar el monto j", se lo define

   $$m(i, j) = \text{menor número de monedas de denominación } d_{i+1}, \ldots, d_n \text{ necesarias para pagar el monto } j$$

   La solución al problema del enunciado se obtiene calculando $m(0, M)$. Se pide,

   a) completar la definición recursiva (backtracking) de $m(i, j)$ considerando los siguientes 4 casos:

   $$m(i,j) = \begin{cases} 0 & j = 0 \\ \infty & j > 0 \wedge i = n \\ m(i+1, j) & j > 0 \wedge i < n \wedge d_{i+1} > j \\ \ldots & j > 0 \wedge i < n \wedge d_{i+1} \leq j \end{cases}$$

   b) dar una solución que utilice programacion dinámica para implementar iterativamente esta solución. Para ello, tener especial cuidado con el orden en que se calculan las celdas de la matriz. Se asume que las denominaciones vienen dadas por un arreglo d de n números naturales.

3. Se considera la siguiente variante del problema de la mochila en que se cuenta con un conjunto de n objetos de peso positivo $w_1, \ldots, w_n$ y valor $v_1, \ldots, v_n$ y dos mochilas de capacidad J y K respectivamente. Se trata de obtener el máximo valor alcanzable cargando ambas mochilas sin exceder sus capacidades. Como no pueden fraccionarse objetos, se requiere escribir un algoritmo que utilice backtracking.

   Para ello, se propone definir $m(i, j, k) =$ "máximo valor alcanzable eligiendo entre objetos $1, \ldots, i$ sin exceder las capacidades j ni k".

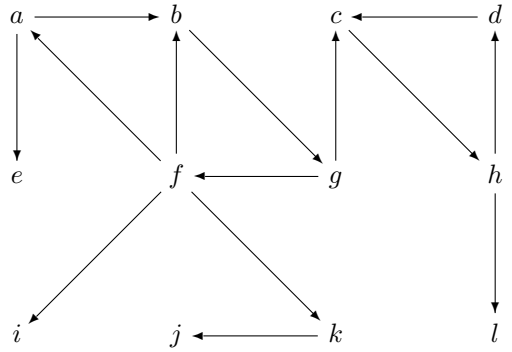
   Puede considerar los siguientes casos:

   $$m(i,j,k) = \begin{cases} \ldots & i = 0 \\ \ldots & i > 0 \wedge j < w_i \wedge k < w_i \\ \ldots & i > 0 \wedge j \geq w_i \wedge k < w_i \\ \ldots & i > 0 \wedge j < w_i \wedge k \geq w_i \\ \ldots & i > 0 \wedge j \geq w_i \wedge k \geq w_i \end{cases}$$

4. En este ejercicio se asume que heap es con máximo (no mínimo) en la raíz. ¿Es el siguiente arreglo un heap? En caso contrario ¿cuál sería el menor número de *movimientos* posibles para convertirlo en un heap? Un *movimiento* es un intercambio entre un hijo y su padre. ¿Cuáles serían esos movimientos? Justifique graficando el heap.

90	80	90	70	80	90	80	70	80	60	70	90	80	80	80	80	60	60	70	60	90

5. Se quiere recorrer el siguiente grafo



a) ¿cuál sería una enumeración posible de los vértices en recorrida DFS asumiendo que el primer vértice que se visita es g?

b) ¿cuál sería una enumeración posible de los vértices en recorrida BFS asumiendo que el primer vértice que se visita es g?