

Diseño de algoritmos

Jesús Bermúdez de Andrés. UPV-EHU
Ejercicios: Algoritmos voraces

Curso 2008-09

1. Considere la siguiente representación con listas de adyacencias del grafo no dirigido $G = (V, A)$. Un par (w, p) en la lista de adyacencia del nodo v significa que existe una arista (v, w) con peso asociado p .

$V = [V(1), V(2), V(3), V(4), V(5), V(6), V(7)]$	
$V(1) = [(3, 3), (4, 10), (5, 11)]$	$V(5) = [(1, 11), (6, 6)]$
$V(2) = [(3, 8), (4, 12), (6, 5), (7, 1)]$	$V(6) = [(2, 5), (4, 2), (5, 6)]$
$V(3) = [(1, 3), (2, 8), (4, 9), (7, 7)]$	$V(7) = [(2, 1), (3, 7), (4, 4)]$
$V(4) = [(1, 10), (2, 12), (3, 9), (6, 2), (7, 4)]$	

- a) Escriba, en orden de selección, las aristas seleccionadas para la solución por el algoritmo de PRIM sobre ese grafo.
 - b) Escriba, en orden de selección, las aristas seleccionadas para la solución por el algoritmo de KRUSKAL sobre ese grafo.
2. Supóngase que un grafo tiene exactamente 2 aristas que tienen el mismo peso.
 - a) ¿Construye el algoritmo de PRIM el mismo árbol de recubrimiento mínimo, independientemente de cuál de esas aristas sea seleccionada antes?
 - b) ¿Y si consideramos el algoritmo de KRUSKAL?
 3. Considérese el siguiente algoritmo para calcular las componentes conexas de un grafo no dirigido $G = (N, A)$ usando la estructura de partición:

```
proc COMPONENTES_CONEXAS( $G = (N, A)$ )  
  for cada nodo  $v \in N$  loop CREA_CONJUNTO( $v$ ) end for  
  for cada arista  $(x, y) \in A$  loop  
    etiquetaX  $\leftarrow$  BUSCAR( $x$ )  
    etiquetaY  $\leftarrow$  BUSCAR( $y$ )  
    if etiquetaX  $\neq$  etiquetaY then  
      UNIR(etiquetaX, etiquetaY)
```

Si $G = (N, A)$ tiene k componentes conexas, ¿cuántas operaciones BUSCAR se realizan? ¿Cuántas operaciones UNIR se realizan? Exprésese el resultado en términos de $n = |N|$, $a = |A|$ y k .

4. Dado un grafo $G = (N, A)$ no dirigido, con pesos asociados a las aristas, el algoritmo KRUSKAL calcula un árbol de recubrimiento mínimo de $G = (N, A)$, empleando para ello la estructura partición. El método comienza ordenando el conjunto de aristas del grafo en orden creciente según sus pesos, para pasar a continuación a tratar una a una estas aristas. Analícese una variante de dicho algoritmo en la cual se emplea un montículo (heap) para almacenar las aristas, de donde se irán cogiendo una a una para ser tratadas de la misma forma que en la versión original. Escribese el algoritmo y calcúlese su orden en el caso peor.
5. El algoritmo DIJKSTRA calcula las distancias mínimas de los caminos desde un nodo origen a cada uno de los nodos restantes, en un grafo dirigido y con pesos no negativos asociados a las aristas.
Escriba una adaptación de ese algoritmo, preservando el orden de complejidad, de manera que:
 - a) calcule, además, el número de caminos distintos de distancia mínima que hay desde el origen a cada nodo y
 - b) calcule lo suficiente para reconstruir cada uno de esos caminos mínimos.
6. Junto con los cursos de verano de la universidad va a celebrarse un ciclo de conferencias. Las conferencias son muchas y cada conferenciante ha decidido la hora de comienzo y de finalización de su conferencia. La dirección de los cursos de verano ha decidido reservar unas cuantas salas para celebrar exclusivamente las conferencias. Escriba y analice un algoritmo que determine cual es el número mínimo de salas que se necesitan.
7. Considere que tenemos n programas para grabar en un disco, pero el espacio de memoria que necesitan excede la capacidad del disco. Cada programa P_i requiere m_i kilobytes de memoria, la capacidad del disco es de C kilobytes y $C < \sum_{i=1}^n m_i$.
 - a) Queremos grabar en el disco el máximo número posible de esos programas. Demuestre si es correcto el siguiente criterio de selección voraz: seleccionar el programa con menor requerimiento de memoria. Si cree que es incorrecto muestre un contraejemplo.
 - b) Se plantea otro problema distinto, si lo que queremos es utilizar la máxima capacidad posible del disco en la grabación de esos programas. Demuestre si para resolver este problema es correcto el criterio de selección voraz siguiente: seleccionar el programa con mayor requerimiento de memoria. Si cree que es incorrecto muestre un contraejemplo.
8. Considere una red de n ordenadores personales que comparte una sola impresora. En un momento dado, las n personas que están trabajando en los ordenadores solicitan simultáneamente la impresión de un documento y se acercan a la impresora para recogerlo.

Diseñe un algoritmo para que la impresora decida el orden en que se deben imprimir los documentos para que la suma de tiempos de espera de las n personas resulte mínimo. El tamaño de cada documento es conocido y la impresora imprime cada documento en un tiempo t_i $i = 1, \dots, n$.