



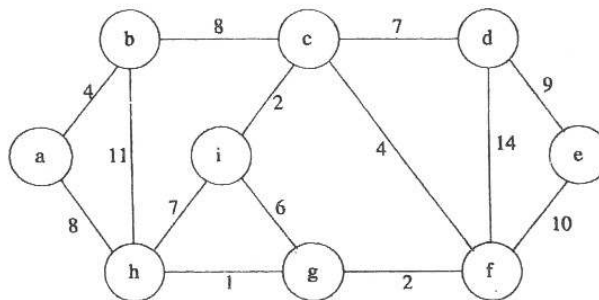
Examen septiembre 2006, preguntas

Algoritmos (Universidade da Coruña)

Examen de Algoritmos
1 de septiembre de 2006

Apellidos:	
Nombre:	Titulación (II / ITIG):

1. (1 punto) Dibuje la matriz de adyacencia y las listas de adyacencia que representan el siguiente grafo no dirigido:



2. (1 punto) A partir de la siguiente estructura de datos para la implementación de conjuntos disjuntos:

```

tipo
  Elemento = entero;
  Conj = entero;
  ConjDisj = vector [1..N] de entero
  
```

y del siguiente pseudocódigo para la unión de dos conjuntos:

```

procedimiento Unir (C, raíz1, raíz2)
  { supone que raíz1 y raíz2 son raíces }
  si raíz1 < raíz2 entonces C[raíz2] := raíz1
  sino C[raíz1] := raíz2
fin procedimiento
  
```

- Escriba el correspondiente pseudocódigo de $\text{Buscar}^*(C, x) : \text{Conj}$, que devuelve el nombre del conjunto (es decir, su representante) de un elemento dado.
 - Indique la complejidad de una secuencia de m búsquedas y $n - 1$ uniones.
3. (1 punto) Calcule, a partir del pseudocódigo que incluirá en su respuesta, la cota exacta para el tiempo de ejecución de la ordenación por selección, indicando las reglas que utiliza.
4. (1,5 puntos) Uso del teorema Divide y Vencerás. Complete una tabla como la siguiente:

algoritmo	caso	caracterización	relación de recurrencia	resultado
BB	medios	ps. no recursiva	$T(n/2) + 1$	$\log n$
SSM	pequeño		$2T(n/2) + cn^4$	$n \cdot \log n$
MS	pequeño		$2T(n/2) + cn^4$	$n \cdot \log n$
QS	mejor	cuando $n \geq 1$	$2T(n/2) + cn^4$	$n \cdot \log n$

Donde las columnas especifican:

- el *caso* en que se utiliza el teorema *Divide y Vencerás* (mejor caso, caso medio, peor, o todos);
- la *caracterización* del caso (¿con qué condiciones se produce, si las hay?);
- la *relación de recurrencia* que corresponde a este caso (sistema de ecuaciones);
- el *resultado* de la aplicación del teorema (complejidad calculada);

y en cada línea se considera un algoritmo diferente:

- a) BB: búsqueda binaria;
- b) SSM: algoritmo recursivo para el problema de la suma de la subsecuencia máxima;
- c) MS: ordenación por fusión;
- d) QS: ordenación rápida.

5. (1,5 puntos) Presente un pseudocódigo del algoritmo de *Kruskal* donde haga uso de un montículo como estructura auxiliar. Determine su complejidad temporal y explique lo que aporta el montículo para mejorar la eficiencia del algoritmo.
6. (1 punto) Construya la tabla con la que podría determinarse en programación dinámica la manera óptima de pagar una cantidad de 17 unidades de valor con un mínimo de monedas, sabiendo que el sistema monetario considerado está constituido por monedas de 1, 2, 8 y 12 unidades de valor. ¿Porqué descartaría el uso de la técnica voraz para resolver este problema?