



## Examen 2014, preguntas

Algoritmos (Universidade da Coruña)

**Examen de Algoritmos**  
**Grado en Ingeniería Informática**  
**Enero de 2014**

<b>Apellidos:</b>	<b>DNI:</b>
<b>Nombre:</b>	

1. (2.5 puntos) Diccionario de datos:

a) Diseñe, escribiendo su pseudocódigo, los algoritmos inicializarTabla, insertar, buscar y eliminar usando exploración cuadrática, de modo que las tres últimas rutinas se ejecuten en un tiempo promedio constante. Use la siguiente declaración de tipos:

**tipo**

ClaseDeEntrada = (legítima, vacía, borrada)

Índice = 0..N-1

Posicion = Índice

Entrada = **registro**

    Elemento : TipoElemento

    Información : ClaseDeEntrada

**fin registro**

TablaDispersión = **vector** [Índice] **de** Entrada

Si utilizase algún procedimiento auxiliar (distinto de la función hash), refleje también su pseudocódigo.

b) Con la siguiente función hash:

hash("a",11) = 8

hash("b",11) = 7

hash("c",11) = 7

hash("d",11) = 7

hash("e",11) = 8

hash("f",11) = 8

muestre el resultado de insertar las claves: "a", "b", "c", "d", "e" y "f" (en ese orden) en la siguiente tabla:

1	2	3	4	5	6	7	8	9	10

cuando se emplea:

1) Exploración lineal

2) Exploración cuadrática

3) Exploración doble usando  $5 - (x \bmod 5)$  como segunda función (siendo x el resultado de la primera función.

Indique asimismo el número total de colisiones que se produce durante las inserciones en cada una de las tres exploraciones.

2. (2 puntos) Compare la ordenación por fusión con la ordenación rápida desde el punto de vista del diseño del algoritmo (técnica de diseño, fases de la técnica utilizada, estrategias para la mejora del tiempo de ejecución...) así como de su complejidad (relaciones de recurrencia según cada caso, que se justificarán y se resolverán, teorema necesario para resolverlas...).

3. (2.5 puntos) Dado el algoritmo siguiente:

```

función Kruskal ( G = (N,A) ) : tipo_salida (1)
  Organizar los candidatos a partir de  $A_i$  (2)
  n := |N|; T := conjunto vacío; peso := 0;
  inicializar n conjuntos, cada uno con un nodo de N;
  repetir
    Seleccionar y extraer un candidato a; (3)
    ConjuntoU := Buscar (a.nodo1); ConjuntoV := Buscar (a.nodo2)
    si ConjuntoU <> ConjuntoV entonces
      Fusionar (ConjuntoU, ConjuntoV);
      T := T U (a);
      peso := peso + a.peso;
    fin si
  hasta |T| = n-1;
  devolver <T, peso>
fin función
  
```

- Precise el tipo de datos de la salida de la función ( $\{1\}$ ).
- Reescriba las instrucciones {2} y {3} introduciendo el uso de un montículo, decisión que justificará en su respuesta.
- Enuncie de forma precisa el problema que resuelve este algoritmo.
- Concrete los elementos característicos de la técnica voraz que correspondan a este ejemplo
- Análisis detallado de la complejidad.

4. (1.5 puntos) Dado el grafo dirigido pesado  $G = (N,A)$ , con  $N = \{a, b, c, \dots, h\}$  y A determinado por la lista de aristas siguiente:

(a,d)	(a,f)	(d,h)	(d,f)	(h,e)	(h,g)	(e,g)	(f,g)	(f,b)	(f,c)	(b,c)
1	2	1	6	2	7	2	3	4	5	2

- Dibuje el grafo G.
- Indique el resultado de un recorrido en profundidad sobre G, partiendo del nodo a.
- Presente una ordenación topológica de los nodos de G.
- Dibuje un árbol expandido mínimo del grafo no dirigido subyacente, indicando igualmente su peso total.
- Dibuje el árbol con los caminos mínimos entre el nodo a y los demás, asociando a cada nodo la distancia mínima calculada.

5. (1.5 puntos) Cálculo de un coeficiente binomial  $C(n,k)$  utilizando la técnica de *Programación Dinámica* de forma que las necesidades de memoria sean mínimas. Proponga un ejemplo concreto, partiendo del triángulo de Pascal, para explicar el algoritmo. Determine su complejidad temporal y espacial.