Metodología de la Programación y Algoritmia

Convocatoria de Junio 2011

Apellidos _____

Nombre _____ DNI _____

1.- El algoritmo de Floyd, cuyo pseudocódigo se indica a continuación, calcula el coste del camino mínimo entre todo par de vértices de un grafo con n vértices.

```
función Floyd(&D[n,n]:real+ U {0}, P[n,n]:real+ U {0})
   i,j,k:natural
   para i←1 hasta n hacer
      para j←1 hasta n hacer
         D[i,j]←P[i,j]
      fpara
   fpara
   para i←1 hasta n hacer
      D[i,i]←0
   fpara
   para k←1 hasta n hacer
      para i←1 hasta n hacer
         para j←1 hasta n hacer
            si D[i,k]+D[k,j] < D[i,j]
               D[i,j] ← D[i,k] + D[k,j]
            fsi
         fpara
      fpara
   fpara
ffunción
```

a.- Indica el tipo de estrategia que sigue y por qué. (0.6 puntos)

b.- Modifica el pseudocódigo del algoritmo para que también almacene la información referente a los vértices que constituyen el camino de coste mínimo entre todos los pares de vértices. Explica por qué se hacen esos cambios.

Una vez realizada la modificación, explica cómo se puede obtener la secuencia de vértices que constituyen el camino de coste mínimo desde un vértice v1 a otro v2. (2.5 puntos)

2.- Para el problema de la Mochila indica cuál es la estrategia más adecuada para resolverlo y **por qué**.

a) Objetos se pueden fraccionar y los pesos son enteros.
b) Objetos se pueden fraccionar y los pesos no son enteros
c) Objetos no se pueden fraccionar y los pesos son enteros
d) Objetos no se pueden fraccionar y los pesos no son enteros

(2.4 puntos)

3.- Describe **detalladamente** qué es la técnica de diseño de algoritmos voraces.

Pon un ejemplo de un problema que se pueda resolver aplicando esta estrategia y resuélvelo para un caso particular explicando por qué sigue la estrategia voraz y los pasos realizados hasta obtener la solución.

(2.0 puntos)

4.- Dado el siguiente algoritmo

```
      función junio2011(&V[x]:entero)
         i, j:natural  n: entero
(1)      para i←1 hasta x
(2)         j ← i
(3)         mientras j>1 hacer
(4)            si V[j]>V[j-1]
(5)               n ← V[j]
(6)               V[j] ← V[j-1]
(7)               V[j-1]← n
(8)            fsi
(9)            j ← j-1
(10)        fmientras
      fpara
   función
```

a.- Calcula la expresión T para los casos peor y mejor aplicando los 3 pasos.

(2.0 puntos)

b.- Calcula la complejidad asintótica y justifica por qué.

(0.5 puntos)

Duración: 3 horas.

Escribe en la parte superior derecha de cada folio tus apellidos, nombre, dni y el nº de folio con respecto al total de folios entregados con el formato: nºfolio / totalfolios (p.ej.: si es el folio 1 de un total de 5 folios: 1/5)