TEMA 9.- ALGORITMOS VORACES

***E 1.-** Escribir un algoritmo voraz para entregar billetes en un cajero automático. Demostrar que es correcto para billetes de 10, 20 y 50 €, y no lo es si no existen billetes de 10 €.*

***E 2.-** Demostrar por inducción que un cajero automático puede entregar cualquier cantidad de dineros en billetes de 20 y 50 €, excepto 10 y 30 Euros. ¿Cuál es la base de inducción correcta en este caso? ¿Qué tipo de inducción hay que utilizar?*

***E 3.-** Demostrar (mediante contraejemplos) que la solución voraz al problema de la mochila no sería correcta si el criterio para seleccionar el siguiente elemento fuese elegir el de más valor o el de menos peso*

***E 4.-** ¿Es aplicable el algoritmo de la mochila continua al problema de la mochila discreta (los elementos no se pueden fraccionar)? ¿Qué tipo de resultados daría?*

***E 5.-** (CLASE) Se quieren almacenar n ficheros de tamaños gi i=1..n en una cinta. El tiempo empleado para recuperar un fichero de la cinta es proporcional a la suma de los tamaños de los ficheros que le preceden y la del propio fichero. Si para recuperar un fichero se ha de rebobinar toda la cinta y lo que se pretende es minimizar el tiempo de recuperación total, encontrar un algoritmo voraz que lleve a término el almacenamiento de los n ficheros y demostrar su corrección.*

***E 6.-** En un sistema operativo, el procesador ha de ejecutar n tareas numeradas de 1 a n. Cada tarea tiene asociado un final de ejecución, t_i, y una ganancia g_i de forma que si la tarea se ejecuta antes de que acabe su periodo de finalización da lugar a esta ganancia y si no la ganancia aportada es cero. Si todas las tareas se ejecutan en tiempo unitario, encuentra un algoritmo voraz que maximice la ganancia aportada por las n tareas. (Ejecución de trabajos con finalización)*

***E 7.-** Considera el problema de encontrar todos los árboles de extensión mínima (si n es más de uno). ¿Cómo se podría modificar el algoritmo de Prim en este caso?. ¿Y para el algoritmo de Kruskal?.*

***E 8.-** Considera el siguiente algoritmo donde Q es un montículo de mínimos que contiene arcos con su peso (el algoritmo utiliza las operaciones genéricas sobre colas de prioridad):*

```
ALGORITMO MST
DATOS:  G = (N,A,p)    // Grafo ponderado, N={1..n}
RES:    T ⊆ A              // Árbol de extensión minimal
AUX:    M ⊆ N
    Q : < Montículo de arcos >
METODO:
    T ← 0; M ← {1}
    Q ← 0M      //Montículo vacío
    Para i ← 1 hasta n hacer UNIR(Q,(i,1))
    Mientras M ≠ N hacer
        (u,v) ← ELEGIR_MINIMO(Q)
        ELIMINAR_MINIMO(Q)
        Si u ∉ M entonces
            T ← T ∪ {(u,v)}
            M ← M ∪ {u}
            Para i ← hasta n hacer UNIR(Q,(i,u))
    FSi
    fMientras
fMST
```

a) ¿Con cuál de los dos algoritmos conocidos (que calculen el árbol de extensión minimal) está relacionado este algoritmo? ¿Por qué?

b) ¿Es correcto el algoritmo? Describe brevemente su funcionamiento.

c) Analiza el algoritmo en el caso peor y relaciónalo con los otros algoritmos estudiados.