ALGORITMIA Y COMPLEJIDAD

LABORATORIO - TEMA 2

EJERCICIO 1

Se tienen n números naturales, siendo n una cantidad par, que tienen que juntarse formando parejas de dos números cada una. A continuación, de cada pareja se obtiene la suma de sus dos componentes, y de todos estos resultados se toma el máximo.

Diseñar un algoritmo voraz que cree las parejas de manera que el valor máximo de las sumas de los números de cada pareja sea lo más pequeño posible, demostrando que la función de selección de candidatos usada proporciona una solución óptima.

Ejemplo:

5	8	1	4	7	9

Vamos a ver un par de formas de resolver el problema (no necesariamente la óptima):

a) Seleccionamos como pareja los elementos consecutivos.
   De esta forma conseguimos las parejas (5, 8), (1, 4) y (7, 9); entonces, al sumar las componentes tenemos los valores 13, 5 y 16, por lo que el resultado final es 16.
b) Seleccionamos como pareja los elementos opuestos en el vector.
   Ahora tenemos las parejas (5, 9), (8, 7) y (1, 4); sumando conseguimos 14, 15 y 5, por lo que el resultado final es 15 (mejor que antes).

¿Habrá un resultado mejor para este problema? ¿Puede generalizarse un método que nos proporcione un algoritmo voraz correcto para cualquier cantidad de datos, y que además sea independiente del valor de los mismos?

EJERCICIO 2

Se tiene que almacenar un conjunto de n ficheros en una cinta magnética (soporte de almacenamiento de recorrido secuencial), donde cada fichero tiene una longitud conocida $l_1, l_2, \dots, l_n$. Para simplificar el problema, puede suponerse que la velocidad de lectura es constante, así como la densidad de información en la cinta.

Se conoce de antemano la tasa de utilización de cada fichero almacenado, es decir, se sabe la cantidad de peticiones p_i correspondiente al fichero i que se van a realizar. Por tanto, el total de peticiones al soporte será:

$$P = \sum_{i=0}^{n} p_i$$

Tras la petición de un fichero, al ser encontrado, la cinta se rebobina automáticamente hasta el principio.

El objetivo es decidir el orden en que los ficheros deben ser almacenados para que se minimice el tiempo medio de carga, creando un algoritmo voraz correcto.

EJERCICIO 3

Se dispone de un vector V formado por n datos, del que se quiere encontrar el elemento mínimo del vector y el elemento máximo del vector. El tipo de los datos no es relevante, pero la comparación entre dos datos para ver cuál es menor es muy costosa, por lo que el algoritmo debe hacer la menor cantidad de comparaciones posible.

Un método trivial consiste en un recorrido lineal del vector para buscar el máximo y el mínimo, lo que requiere un total de aproximadamente $2n$ comparaciones. Este método no es lo suficientemente rápido, por lo que se pide implementar un método voraz que realice un máximo de $3n/2$ comparaciones.

EJERCICIO 4

En un país, hay varias ciudades que necesitan estar conectadas mediante una red de fibra óptica para mejorar las comunicaciones. Cada tramo de fibra óptica entre dos ciudades tiene un coste específico de instalación en euros. El objetivo es conectar todas las ciudades de manera que el coste total de instalación sea el menor posible.

El objetivo consiste en diseñar un algoritmo que conecte todas las ciudades, minimizando el coste total de instalación. Para ello, partimos de una lista de todas las ciudades y, una lista de posibles tramos de fibra óptica entre las ciudades con sus respectivos costes de instalación.

Restricciones:

- Cada ciudad debe estar conectada a la red de fibra óptica.
- La red debe ser continua, es decir, no debe haber ciudades aisladas.

EJERCICIO 5

En una red de computadoras, cada computadora está conectada a otras mediante cables de red. Cada cable tiene un tiempo de transmisión específico en milisegundos. Un administrador de red necesita encontrar la ruta más rápida para enviar datos desde un servidor principal al resto de computadoras de la red.

Diseñe un algoritmo que calcule la ruta más rápida desde el servidor principal al resto de computadoras. Para ello disponemos de una lista de computadoras y otra lista con los cables de red disponibles y sus respectivos tiempos de transmisión.

EJERCICIO 6

Dado un conjunto de reservas de pistas de pádel, cada una con una hora de inicio y otra de finalización, diseñe un algoritmo que calcule el número mínimo de pistas necesarias para que todas las reservas puedan llevarse a cabo sin conflictos de horario.

Para ello se dispone de una lista de intervalos de tiempo, donde cada intervalo está representado por una hora de inicio y una de finalización de la reserva.

Ejemplo:

Entrada → `[(10, 12), (9, 11), (11, 13)]`
Salida → 2

- La reserva de 9 a 11 ocupa una pista.
- La segunda reserva de 10 a 12 necesita una nueva pista porque se solapa con la primera.
- La tercera reserva de 11 a 13 puede utilizar la misma pista que la primera después de que esta termine.

Entregables: un ejercicio a elegir entre los problemas 2 y 3, un ejercicio a elegir entre los problemas 4 y 5, y el problema 6.