3er Parcial

1. Leer un archivo de una cinta tiene diferente penalidad que a leer un archivo de un disco, ya que hay que adelantar la cinta hasta la posición donde leer el archivo. Suponiendo que existen n archivos, donde el tamaño del archivo i es el natural l_i, deseamos obtener una permutación de $\langle 1, 2, \ldots, n\rangle$ que minimice el costo de lectura promedio.

Si tenemos una permutación $\langle \pi_1, \pi_2, \ldots, \pi_n\rangle$, entonces el costo de leer el archivo k es

$$costo(k) = \sum_{i=1}^{k} l_{\pi_i}$$

y el costo promedio es

$$\sum_{i=1}^{n} \frac{costo(i)}{n}$$

.

a) Dar una estrategia greedy que resuelva el problema de minimizar el costo de lectura promedio.

b) Probar que esta estrategia nos da efectivamente un óptimo.

2. En la práctica se ha visto el problema de cambio. Dado un importe D, e ilimitadas monedas de denominación M_i $1 \leq i \leq k$, se desea encontrar la mínima cantidad de monedas necesarias para formar D.
En este ejercicio utilizaremos la programación dinámica para dar una solución al problema. La eureka de este ejercicio consiste en llamar $M(l, j)$ al subproblema de obtener el importe j utilizando las denominaciones M_i $1 \leq i \leq l$.

a) Dar una definición por casos de $M(l, j)$ que sirva para resolver el problema por dinámica.

b) Escribir una función en C que resuelva de manera dinámica bottom–up cuál es la mínima cantidad de monedas necesarias para formar un importe dado.

c) Analizar la complejidad del algoritmo propuesto.

3. Para el algoritmo *quicksort* utilizando como pivot el primer elemento:

a) Describir su funcionamiento.

b) Describir alguna forma de los arreglos en que se da el peor caso de complejidad.