

Exercici 2 (3 punts) – Copistes –

Abans de la invenció de la impremta, era molt difícil fer una còpia d'un llibre. Tots els continguts havien de ser redactats a mà pels anomenats copistes. A un copista se li donava un llibre i , després de diversos mesos de treball, tornava una còpia del mateix. El temps que trigava era, molt probablement, proporcional al nombre de pàgines del llibre. La feina devia ser prou avorrida i per això podríem assumir que tots els copistes d'un monestir trigaven el mateix temps a copiar una pàgina.

El monestir de Pedralbes va decidir fer una còpia dels llibres de la seva biblioteca i, donat que no tenen copistes propis, han d'enviar els llibres a un altre monestir que sí que en tingui.

Podem assumir que hi ha un total de n llibres per copiar i que el llibre i -èsim té p_i pàgines. A més, els llibres tenen un ordre predeterminat. Una vegada triat el monestir on es faran les còpies, s'hauran de repartir els llibres entre els seus m copistes. Cada llibre només se li pot assignar a un copista, i a cada copista només se li pot assignar una seqüència contigua de llibres (d'acord amb l'ordre inicial dels llibres). Amb aquesta forma d'assignar llibres a copistes es minimitza el temps de buidar i tornar a omplir la biblioteca. El temps necessari per fer la còpia total de la biblioteca queda determinat pel temps que necessita el darrer copista que finalitza la còpia dels llibres que se li han assignat.

El que no tenen molt clar els encarregats de la biblioteca és com fer l'assignació de llibres a copistes per garantir que el temps de còpia total de la biblioteca sigui el més curt possible. Ajudeu a aquests monjos i dissenyeu un algorisme eficient per a trobar l'assignació òptima dels n llibres als m copistes del monestir. Podem assumir que coneixeu el temps t_p que necessita cadascun dels copistes per a copiar una pàgina.

Solució:

Todos los copistas copian a la misma velocidad, asumo que $t_p = 1$ y multiplicaremos por t_p al final.

Los libros no se pueden desordenar, una asignación a un copista será de todos los libros entre el i y el j , para $i \leq j$ adecuados.

Una solución óptima del problema es una asignación de un segmento inicial de libros a un copista seguida de una asignación óptima de los libros restantes a $m - 1$ copistes. El coste es el máximo de los dos valores.

Vamos a calcular $T[i][k]$, el tiempo mínimo para copiar los libros i, \dots, n cuando disponemos de k copistas.

Utilizaremos $P[i][j]$, $1 \leq i \leq j \leq n$ que nos indica el número de páginas de los libros i a j .

A partir de la estructura de suboptimalidad obtenemos la siguiente recurrencia:

$$T[i][k] = \begin{cases} P[i][n] & k=1, \\ P[n][n] & i=n, \\ \min_{i < j \leq n} \{ \max\{P[i][j], T[j][k-1]\} \} & \end{cases}$$

Podemos precalcular P en tiempo $O(n^2)$ o precalcular $Q[i]$ el número de páginas de los libros 1 a i en tiempo $O(n)$. Así podemos utilizar $Q[j] - Q[i]$ en vez de $P[i][j]$, si $i < j$, y ahorrar algo de espacio y tiempo.

Para obtener la solución utilizaremos PD, calculando la tabla T de acuerdo con la recurrencia, empezando por la ultima columna. Como tenemos que obtener la solución, tambien guardaremos en una tabla auxiliar, para cada valor $[i][k]$ el valor j dónde se alcanza el mínimo. Si precalculamos P , utilizando PD podemos obtener T en tiempo $O(n)$ por elemento, lo que nos da un coste de $O(n^2m)$ para todo el algoritmo.