

Algoritmos y Estructuras de Datos III, TP2

Nicolás Chehebar, Matías Duran, Lucas Somacal

Índice

1. Problema 1	2
1.1. El Problema	2
1.1.1. Descripción	2
1.1.2. Ejemplos	2
1.2. El Algoritmo sin podas	2
1.2.1. Resumen	2
1.3. Complejidad	2
1.4. Experimentación	2
1.4.1. Contexto	2
1.5. Conclusiones	2
2. Problema 2.1	2
3. Problema 2.2	2
4. Problema 3	2

1. Problema 1

1.1. El Problema

1.1.1. Descripción

Planteado de otra forma, el problema a resolver consiste en una situación en la que tenemos n trabajos t_1, t_2, \dots, t_n y dada cualquier división de los trabajos en dos secuencias $A = (t_{a_1}, t_{a_2}, \dots, t_{a_{|A|}})$ y $B = (t_{b_1}, t_{b_2}, \dots, t_{b_{|B|}})$ con $a_i < a_j \wedge b_i < b_j \implies i < j$ (cada secuencia representa las impresiones que realizó una impresora) tiene asociado un costo; donde este viene dado por la suma del costo de A y el de B . El costo de A es $\sum_{i=1}^{|A|} \text{costo}(t_{a_i}, t_{a_{i-1}})$ donde costo es una función que toma valores en \mathbb{N}_0 y $\text{costo}(t_i, t_j)$ está definido si $i > j$ con $i \in [1, 2, \dots, n] \wedge j \in [0, 1, \dots, n-1]$ y representa el costo de poner el trabajo i sobre el j (el costo de poner sobre el trabajo t_0 es el de ponerlo sobre la máquina vacía y $a_0 = 0$). El costo de B se calcula análogamente.

El problema pide dados los trabajos y la función de costo, dar A o B que minimice el costo y decir cuánto es este costo (basta dar uno de los dos ya que el otro se deduce por ser el complemento -en el conjunto de trabajos-).

1.1.2. Ejemplos

- En el caso en que la entrada es

4
2
300 3
300 3 3
300 3 3 3

 tenemos 4 trabajos que sacando el primero son excesivamente caros de poner por primera vez en una máquina, luego si ponemos todos en la misma, el costo será $2 + 3 + 3 + 3 = 11$ y una máquina tendrá todos los trabajos (si todos no están en la misma, en algún momento pagamos 300 y el costo ya sería mayor a 11).
-

1.2. El Algoritmo sin podas

1.2.1. Resumen

1.3. Complejidad

1.4. Experimentación

1.4.1. Contexto

La experimentación se realizó toda en la misma computadora, cuyo procesador era Intel(R) Atom(TM) CPU N2600 @ 1.60GHz, de 36 bits physical, 48 bits virtual, con una memoria RAM de 2048 MB. Para experimentar, se calculó el tiempo que tardaba el algoritmo sin considerar el tiempo de lectura y escritura ni el tiempo que llevaba armar la matriz (ya que se leía un dato, se escribía la matriz y luego se leía el siguiente). El tiempo se medía no como tiempo global sino como tiempo de proceso, calculando la cantidad de ticks del reloj (con el tipo `clock_t` de C++) y luego se dividía el delta de ticks sobre `CLOCKS_PER_SEC`. En todos los experimentos el tiempo se mide en segundos.

1.5. Conclusiones

2. Problema 2.1

3. Problema 2.2

4. Problema 3