MA3705 Algoritmos Combinatoriales.
Profesor: Iván Rapaport.
Auxiliares: Antonia Labarca y Cristian Palma.

Auxiliar 5

Suponga que cuenta con un procesador *monocore* (es decir, que solo puede realizar una tarea a la vez). Además, tiene un conjunto finito $T = \{a_1, a_2, \dots, a_n\}$ de posibles tareas a realizar, todas con duración de 1 unidad de tiempo. Cada una de las tareas tiene asociado un plazo d_i tal que $\forall i \in [n], 1 \leq d_i \leq n$, tal que si la tarea a_i se concluye a lo más en el tiempo d_i se recibe una ganancia w_i.

Diremos que un *schedule* S de T es un ordenamiento de todas las tareas de T (note que hay $n!$ *schedules* posibles). Se considerará que la primera tarea inicia en el tiempo 0 y termina en el tiempo 1, la segunda inicia en el tiempo 1 y termina en el tiempo 2, ..., la n-ésima inicia en el tiempo $n-1$ y termina en el tiempo n.

Decimos que para un determinado *schedule* S, una tarea a_i está **a tiempo** si está en la posición a lo más d_i del ordenamiento (por lo que se completaría a lo más en el tiempo d_i). Por el contrario, está **atrasada** si está en una posición posterior.

Sean $E(S)$ el conjunto de las tareas a tiempo y $L(S)$ el conjunto de las tareas atrasadas. Diremos que un *schedule* está en **forma normal** si para algún $0 \leq j \leq n$ las primeras j tareas están a tiempo y las siguientes $n-j$ están atrasadas. Diremos que está en **forma canónica** si está en forma normal y además las tareas a tiempo están ordenadas ascendentemente según su deadline.

P1 Justifique que para S se puede construir $\bar{S}$ en forma canónica de forma que $E(S) = E(\bar{S}), L(S) = L(\bar{S})$.

Decimos que un conjunto I de tareas es independiente si existe un *schedule* S tal que $I \subseteq E(S)$. Sea $\mathcal{I} = \{A \subseteq T | A \text{ es independiente}\}$.

Para $0 \leq t \leq n, A \subseteq T$, sea $N_t(A) = |\{a_i \in A | d_i \leq t\}|$.

P2 Pruebe que para $A \subseteq T$ son equivalentes

a) $A \in \mathcal{I}$

b) $\forall t = 0 \dots n, N_t(A) \leq t$

c) Si se ordena a ascendentemente según el plazo de cada tarea, ninguna tarea estará atrasada.

P3 Se define el sistema de independencia $(T, \mathcal{I})$. Pruebe que es una matroide.

P4 Diseñe y analice un algoritmo *greedy* que resuelva el problema.