

# **75.29 Teoría de Algoritmos I**

Trabajo Práctico Nº 3

**Fecha de Entrega: 05/12/2014**

## PLANIFICACIÓN DE ORDENES DE TRABAJO

Dado un conjunto de  $n$  trabajos que hay que realizar en una máquina determinada, tales que cada trabajo  $a_i$  requiere  $t_i$  tiempo de máquina para ser completado y produce un beneficio  $b_i$  y tiene asociado un vencimiento  $v_i$

Y considerando que un trabajo no puede interrumpirse una vez que se inició, y la máquina solo puede ejecutar un único trabajo a la vez y el beneficio  $b_i$  se percibe solo si el trabajo  $a_i$  se completa antes de su vencimiento  $v_i$  (caso contrario el beneficio es 0).

Dados los siguientes escenarios:

**a) Todos los tiempos  $t_i$  son enteros entre 1 y  $n$ . Los vencimientos  $v_i$  también son enteros.**

Escribir un programa que encuentre la planificación buscada en tiempo polinomial (programación dinámica). Calcular el orden de la solución encontrada.

Los datos vienen dado en un archivo de texto donde cada linea contiene la tupla (valores separados por coma):

$t_i, b_i, v_i$

por cada trabajo.

La salida es una secuencia de enteros, ordenados según el orden en que deben ejecutarse cada uno de los trabajos.

**b) Los tiempos  $t_i$  y los vencimientos  $v_i$  son reales (arbitrarios)**

**¿Existe una planificación tal que todos los trabajos se completan y el beneficio total es de al menos  $K$ ?**

Demostrar que responder a la pregunta enunciada es NP-completo.

Calcular el orden del algoritmo verificador (que dada una planificación comprueba si es solución o no)

Reducir un problema NP-completo al problema bajo estudio.

## INFORME:

Se debe presentar un informe impreso con el análisis, diseño y código fuente de todos los algoritmos desarrollados (incluyendo funciones o métodos auxiliares para leer archivos, etc.), se debe incluir diagramas de clase y cálculo de órdenes de todos los algoritmos implementados. Se deberá enviar por mail a [mfranzo+tda2014+tp3@gmail.com](mailto:mfranzo+tda2014+tp3@gmail.com) el informe anterior en formato .pdf, el código fuente con archivo para compilación automática (Makefile, build.xml, etc.) una versión compilada para linux, un readme con instrucciones de uso y varios archivos .txt con datos de prueba.

Todos los programas desarrollados se tienen que poder ejecutar en las máquinas de la facultad (con Linux)