75.29 - Teoría de Algoritmos I: Trabajo Práctico n. 2

Equipo Q:
Lavandeira, Lucas (#98042)

Rozanec, Matias (#97404)
rozanecm@gmail.com

Sbruzzi, José (#97452)

14.mayo.2018

Facultad de Ingeniería, Universidad de Buenos Aires

Índice

Parte I
Resolución

1. Parte 1: Spy vs Spy

2. Parte 2

2.1. Ejercicio 1

2.2. Ejercicio 2

2.3. Ejercicio 3

Como primer paso, demostraremos que, dada una cantidad n de cursos, si llamamos m a la cantidad máxima de cursos superpuestos en cualquier instante de tiempo, la cantidad de aulas necesarias para realizar la asignación es m.

Demostración
Para demostrar esto, se plantearán las cotas superior e inferior, y se demostrará que ambas coinciden, siendo el valor de ambas: m.

Cota inferior
De haber m cursos que se dictan al mismo tiempo, como no se pueden dictar cursos distintos en una misma aula en simultáneo, queda probado que como mínimo debe haber tantas aulas como cursos se dictan en simultáneo. Por lo tanto, es válido afirmar que

$$Aulas\ necesarias \geq m$$

Cota superior
Para todo instante t de tiempo habrá, según lo demostrado anteriormente, como mínimo m aulas disponibles. Se puede comprobar rápidamente que para ningún t se necesitarán más que m aulas, dado que si dos cursos no se superponen temporalmente, no hay razón por la que no puedan compartir una misma aula. Además se está tratando el caso en que todas las aulas son iguales, lo que evita complicaciones más allá del análisis presentado.

Llegamos entonces a que

$$Aulas\ necesarias \leq m$$

Queda así demostrado que $Aulas\ necesarias = m$

2.3.1. Pseudocódigo

Algorithm 1: Pseudo código que resuelve el problema.

Data: Horarios de inicio y finalización de cada uno de los n cursos: $T_{inicio,j}$ y $T_{fin,j}$ denotan los tiempos de inicio y finalización del curso $j-$ésimo. Conjunto de todos los tiempos: $T_i, i \in (1, 2n)$

Result: Menor cantidad de aulas necesarias para acomodar todos los cursos suponiendo que todas las aulas son iguales.

```
min time slice ← ∞;
min time ← ∞;
max time ← 0;
foreach T_i do
    if (current min slice = |T_i − T_j|) < min_aulas, i ≠ j then
        min time slice := current min;
    if T_i < min time then
        min time := T_i;
    if T_i > max time then
        max time := T_i;
current time := min time;
max superpositions := 0;
while min time < max time do
    current num of superpositions := 0;
    foreach T_inicio,j , T_fin,j do
        if T_inicio,j ≥ current time slice AND > T_fin,j then
            ++current num of superpositions;
    if current num of superpositions > max superpositions then
        max superpositions := current num of superpositions;
    current time += min time slice;
```