

1. Selección de actividades

1.1. Introducción

El objetivo de la práctica es implementar una estrategia óptima para resolver el problema de la selección de actividades mediante un algoritmo voraz. El problema es tratado en profundidad en el apartado 16.1 del libro “Introduction to Algorithms” (Cormen et. al., 3ª edición).

1.2. Enunciado del problema

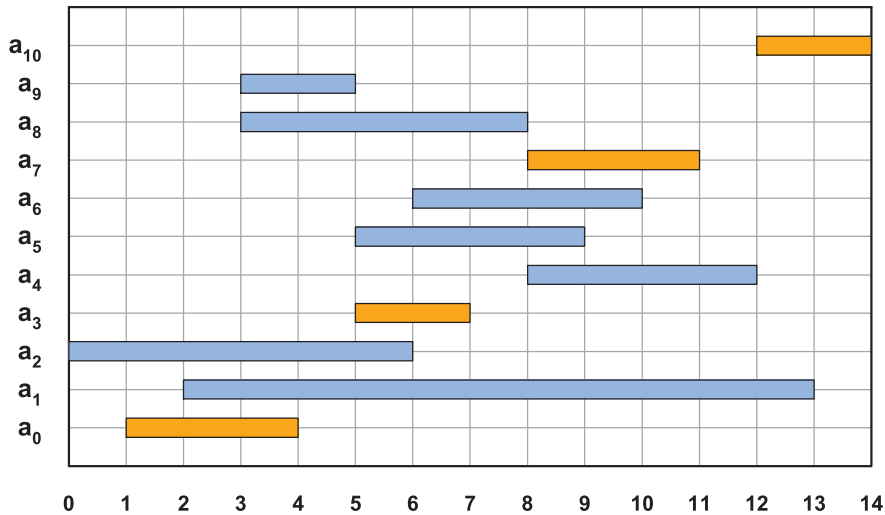
Sea un conjunto A de n actividades $\{a_0, a_1, \dots, a_{n-1}\}$ que necesitan utilizar un recurso común, por ejemplo, una sala de reuniones. El recurso solo puede ser usado por una actividad en cada momento. Cada actividad tiene un instante de comienzo c_i y un instante de finalización f_i , donde $0 \leq c_i < f_i < \infty$.

Si se selecciona la actividad a_i , se desarrolla en el intervalo semiabierto de tiempo $[c_i, f_i)$. Las actividades a_i y a_j son compatibles si sus intervalos $[c_i, f_i)$ y $[c_j, f_j)$ no se solapan, es decir, si $c_i \geq f_j$ o $c_j \geq f_i$.

El *problema de selección de actividades* consiste en determinar un subconjunto de actividades compatibles cuya cardinalidad sea máxima.

Por ejemplo, sea el siguiente conjunto de actividades:

i	0	1	2	3	4	5	6	7	8	9	10
c_i	1	2	0	5	8	5	6	8	3	3	12
f_i	4	13	6	7	12	9	10	11	8	5	14



Un subconjunto S de actividades compatibles es $\{a_2, a_4, a_{10}\}$. Sin embargo, no es un subconjunto de cardinalidad máxima, como lo son $\{a_0, a_3, a_7, a_{10}\}$ (en naranja en la figura) y $\{a_9, a_3, a_4, a_{10}\}$ (ya que tienen 4 elementos).

1.2.1. Descripción de la entrada

En su primera línea la entrada contiene el número de tareas, que será un número entero positivo $n \in [1, 500]$. En la línea siguiente se especificarán los tiempos de comienzo de las tareas c_i , para $i = 1, \dots, n$, separados por espacios en blanco. En la tercera línea se indicarán los tiempos de finalización de las tareas f_i , para $i = 1, \dots, n$, separados por espacios en blanco. Se asume que las entradas cumplen las precondiciones del problema: $0 \leq c_i < f_i < \infty$.

1.2.2. Descripción de la salida

La salida contendrá el número máximo de tareas que pueden llevarse a cabo sin solaparse en el tiempo (no se pide indicar qué tareas se realizan), seguido de un salto de línea.

1.2.3. Ejemplo de entrada

```
11↵
1 2 0 5 8 5 6 8 3 3 12↵
4 13 6 7 12 9 10 11 8 5 14↵
```

1.2.4. Salida para el ejemplo de entrada

4