Técnicas de diseño de algoritmos
Algoritmos voraces
Ejercicios (Bloque 2): con pautas de resolución

Luis Javier Rodríguez Fuentes
Amparo Varona Fernández

Departamento de Electricidad y Electrónica
Facultad de Ciencia y Tecnología, UPV/EHU
luisjavier.rodriguez@ehu.es
amparo.varona@ehu.es

OpenCourseWare 2015
Campus Virtual UPV/EHU

Algoritmos voraces – Ejercicios (Bloque 2)

(B2.1) Un coche necesita repostar cada n kilómetros y su consumo es directamente proporcional a la distancia recorrida. El coche ha de recorrer una ruta de D kilómetros entre dos ciudades A y B. En la ruta hay $m+1$ gasolineras, la primera (de índice 0) en A y la última (de índice m) en B. La distancia d_i entre dos gasolineras consecutivas $i-1$ e i se almacena en la lista $d = [d_1, d_2, \ldots, d_m]$. Escribir en lenguaje Python una función que, siguiendo un esquema voraz, determine el número mínimo de paradas necesario para hacer el recorrido. Supóngase que $d_i \leq n \;\; \forall i \in [1, m]$. **Importante:** el criterio voraz consistirá en parar únicamente cuando la distancia a la siguiente gasolinera sea mayor que la distancia que puede recorrer el coche con la gasolina que le queda. Demostrar que este criterio conduce a la solución óptima.

La demostración de optimalidad se basará en suponer que existe otra solución mejor, que coincide con la solución voraz hasta la parada k, observando que la parada $k+1$ de la solución presuntamente óptima será anterior a la parada $k+1$ de la solución voraz, pues es distinta y no puede ser posterior.

Algoritmos voraces – Ejercicios (Bloque 2)

(B2.2) Se han planificado n actividades, de las cuales se conoce la hora de inicio (intervalo cerrado) y la hora de terminación (intervalo abierto): actividad $i \rightarrow [t_i^{\text{ini}}, t_i^{\text{fin}})$, y se dispone de m salas, con $m \geq n$. Se desea diseñar un calendario tal que se realicen **todas las actividades**, sin que se solapen entre sí, **en el menor número de salas posible**. Escribir en lenguaje Python una función que, aplicando un esquema voraz, obtenga la solución óptima al problema planteado. **Importante:** se ordenarán las actividades por hora de comienzo y se irá escogiendo para cada actividad la primera sala que no tenga ocupada dicha hora, o bien, si todas las salas abiertas estuvieran ocupadas, se abrirá una nueva sala. Demostrar que este criterio conduce a la solución óptima.

Nota: Este problema equivale al de colorear un grafo con el mínimo número de colores, sin más que asociar a cada actividad un nodo y conectar mediante un arco los nodos que son incompatibles (es decir, los que corresponden a actividades que se solapan en el tiempo).

La demostración de optimalidad se hará por inducción sobre el número de actividades (n).