

## RELACIÓN DE PROBLEMAS Algoritmos Voraces

### Problema 1: Planificación de tareas con tiempo límite

El repartidor de Uber tiene que entregar N pedidos de diferente valor teniendo que hacerlo antes de lo indicado en la tabla. En un instante determinado solamente puede entregar un pedido.

Pedido	1	2	3	4	5	6	7	8	9
Tiempo límite	1	5	5	6	6	4	4	2	2
A cobrar	60	70	80	20	20	30	50	50	90

Si entrega tarde un pedido, al cliente le sale gratis. Construye una heurística de tu invención y un algoritmo Voraz basado en lo explicado en el libro de G. Brassard, P. Bratley en la sección de "*Planificación con plazo fijo*" (página 233 y siguientes, en la edición de 1997), que le indique el orden que debe seguir en el reparto para obtener el máximo beneficio. Aplícalo al ejemplo detallando TODOS los pasos.

### Problema 2: Reordenación de vector con suma máxima

Sean los vectores A y B de n números naturales cada uno, el primero de los cuales, A, está ordenado crecientemente. Implementa un algoritmo voraz que obtenga una reordenación del vector B tal que maximice la suma de los elementos de A que no superan al elemento correspondiente de B es decir, maximiza.

$$S = \sum_{i=1, A(i) \leq B(i)}^n A(i)$$

Ejemplo: A = {1, 4, 6, 7, 9}, B = {3, 2, 4, 6, 5}

Una posible reordenación de B (no la óptima) podría ser: B = {2, 4, 5, 3, 6}  
Obtendremos un valor de  $S = 1 + 4 = 5$

¿Cuál sería la solución óptima para este ejemplo?

### Problema 3: Red de ordenadores

Una empresa pleistocénica dispone de 10 ordenadores conectados todos entre sí por una tupida red de conexiones punto a punto. Cada conexión.  $C(i, j)$  entre el nodo  $i$  y  $j$  (con  $i, j \in \{1..10\}$ ) tiene un coste asignado que sigue la fórmula  $C(i, j) = (i + j) \% 8$ . Tras la contratación de un ingeniero en informática de la Universidad de Jaén, la empresa consiente la modernización de sus conexiones asegurando la conectividad de su red de nodos y minimizando el coste. Diseñar un algoritmo voraz que resuelva el problema y aplicarlo a los datos del enunciado.

	1	2	3	4	5	6	7	8	9	10
1	-	3	4	5	6	7	0	1	2	3
2	3	-	5	6	7	0	1	2	3	4
3	4	5	-	7	0	1	2	3	4	5
4	5	6	7	-	1	2	3	4	5	6
5	6	7	0	1	-	3	4	5	6	7
6	7	0	1	2	3	-	5	6	7	0
7	0	1	2	3	4	5	-	7	0	1
8	1	2	3	4	5	6	7	-	1	2
9	2	3	4	5	6	7	0	1	-	3
10	3	4	5	6	7	0	1	2	3	-

### Problema 4: Iluminación continua

Una sala ha de estar constantemente iluminada por un mínimo de  $M$  bombillas. Para ello, disponemos de un conjunto finito de  $N$  bombillas. Cada bombilla  $i$  tiene un tiempo de duración  $t_i$  días. Una vez conectada, no se puede apagar, continuando encendida hasta que se gaste. Se trata de maximizar el número de días que podemos tener la sala iluminada. Diseñe una función basada en una heurística voraz que devuelva el orden en el que se tienen que ir encendiendo las bombillas.

*Ejemplo: Pongamos una heurística de selección aleatoria. Es mala, por eso se pone como ejemplo, no se puede usar.*

Bombilla	1	2	3	4	5	6	7	8	9	10
Duración	1	4	2	3	1	1	3	1	5	2

Orden (M=3)	6	9	3	4	7	2	1	8	10	5
-------------	---	---	---	---	---	---	---	---	----	---

*Con la elección de 3 bombillas necesarias y la selección aleatoria sale un total de 7 días.*

1	2	3	4	5	6	7	8
6, 9, 3	4, 9, 3	4, 9, 7	4, 9, 7	2, 9, 7	2, 1, 8	2, 10, 5	2, 10

## Problema 5: El camionero de Jaén

Un camionero conduce desde París a Jaén siguiendo una ruta dada y llevando un camión que le permite, con el tanque de gasolina lleno, recorrer  $n$  kilómetros sin parar. El camionero dispone de un mapa de carreteras que le indica las distancias entre las gasolineras que hay en su ruta. Como va con prisa, el camionero desea pararse a repostar el menor número de veces posible. Diseñar un algoritmo voraz para determinar en qué gasolineras tiene que parar y demostrar que el algoritmo encuentra siempre la solución óptima

## Problema 6: Distribución de libros

Se quiere colocar una colección de  $n$  libros en una estantería de  $m$  baldas. Los libros son muy dispares y sus pesos difieren mucho unos de otros. Se quiere repartir el peso lo más uniformemente posible, evitando así que se comben. Diseñar una heurística voraz que resuelva el problema, *asignando en cada iteración un libro a la balda que menor peso está soportando*.