

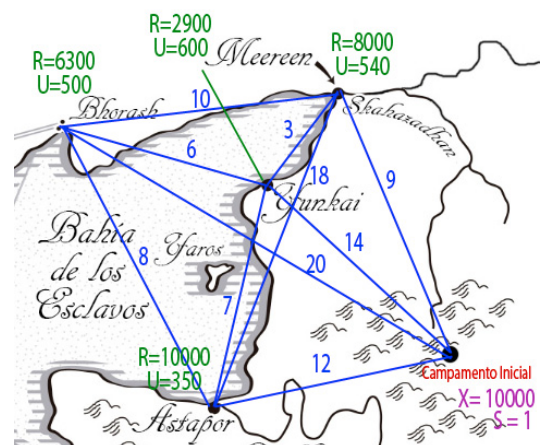
1. **Jon Nieve**, ascendido a oficial, está al mando de un grupo de **N** soldados ordenados por edad. Ha ideado una descabellada formación militar que consiste en defender en fila india siguiendo dicho orden. Su objetivo es mentir sobre su edad para colocarse en la posición central de la fila (la posición más segura en una posible batalla). Sin embargo la estrategia no es el único punto débil de Jon Nieve, la algoritmia también y Jon tiene un problema. Ha llegado un nuevo grupo de **M** soldados que también tenemos ordenados por edad. Jon Nieve necesita saber qué edad debe decir que tiene para colocarse en la posición central de la fila ordenada de soldados que resultaría si se mezclasen ambos grupos. La complejidad debe ser estrictamente menor de **$O(N+M+1)$** .

2. **Daenerys**, noble heredera de la casa Targaryen, se ha propuesto recuperar el poder en los siete reinos y para ello debe conseguir el apoyo de **N** pueblos guerreros de la región del Sur. Todos y cada uno de los pueblos están unidos por áridos caminos y peligrosas rutas marítimas, de los que Daenerys conoce el tiempo en días necesario para recorrerlos. (por cada par de pueblos **i,j** se conoce el tiempo **T_{ij}** necesario para llegar de **i** hasta **j**). Daenerys tiene un innato poder de convicción (véase, tiene unos temibles dragones) que le permite obtener una cantidad determinada de oro (**R_i**) y unas unidades de ejército (**U_i**) de cada uno de los pueblos. Sin embargo, cada día debe gastar una cantidad concreta de oro **S** en víveres por cada unidad de ejército a sus órdenes. Daenerys comienza su aventura desde su campamento inicial, con una cantidad de **X** monedas de oro y sin ejército a sus órdenes. En todos los casos debe también acabar la aventura en su campamento.
 - **Variante 1:** Daenerys necesita saber en qué orden debe visitar todos y cada uno de los pueblos tardando el menor tiempo posible y sin quedarse sin oro para pagar a su ejército.
 - **Variante 2:** Daenerys necesita saber en qué orden debe visitar todos y cada uno de los pueblos para maximizar el oro tras conseguir todas las unidades de ejército disponibles.
 - **Variante 3:** La codicia ha podido con Daenerys y su objetivo es obtener la mayor cantidad de oro sin importarle el tiempo ni las unidades de ejércitos conseguidas, ya que cuando reúna lo máximo posible, se fugará al paraíso fiscal de las islas de hierro. En este caso no tiene por qué visitar todos los pueblos; eso sí, cada vez que recauda oro de un pueblo y decida visitar otro, debe llevarse su ejército al siguiente pueblo (y pagar sus víveres) para no levantar sospechas.

Resolver las tres variantes del problema utilizando un algoritmo voraz, programación dinámica y backtracking.

Ejemplo (no tiene porque reflejar el caso óptimo para ninguna de las variantes, es simplemente una posible solución):

| Día | Posición | Unidades | Oro Obtenido | Oro Gastado* | Balance |
|-----|------------|----------|--------------|------------------------|---------|
| 0 | Campamento | 0 | 0 | 0 | 10000 |
| 12 | Astapor | 350 | 10000 | 0 | 20000 |
| 20 | Bhorash | 850 | 6300 | $350 \cdot 8 = 2800$ | 23500 |
| 26 | Yunkai | 1450 | 2900 | $850 \cdot 6 = 5100$ | 21300 |
| 29 | Meereen | 1990 | 8000 | $1450 \cdot 3 = 4350$ | 24950 |
| 38 | Campamento | 1990 | 0 | $1990 \cdot 9 = 17910$ | 7040 |



* Refleja el gasto en víveres de mover las unidades acumuladas del poblado anterior al actual, con $S=1$ (cada unidad consume 1 de oro al día) multiplicado por el número de días de viaje entre poblados.

Nota 1: Ignore el tiempo invertido en cada poblado para negociar el oro y ejército obtenidos. Daenerys, sorprendentemente, tampoco se dejará engañar ni se dejará secuestrar en ninguno de los poblados. Ignore también los víveres que pueda consumir Daenerys.

Nota 2: Resuelva el problema con los datos que se muestran en el ejemplo y con el fichero de datos proporcionado en Campus Virtual.

3. **Juegos.** Los Lannister siempre pagan sus deudas y Tyrion Lannister ha prometido a su hermano Jaime repartir el tesoro encontrado formado por N objetos de determinado valor V_i y peso P_i . Sobre una mesa coloca la colección de N objetos. Deberán coger alternativamente un objeto de la colección. El peso total de los objetos que coge un jugador no puede sobrepasar una cantidad dada K (no pueden transportar más). El objetivo para ambos es lograr reunir la colección de objetos de mayor valor. Construya un algoritmo que dada una configuración del juego la evalúe, y devuelva la decisión que produce el resultado más favorable. Opcionalmente, implemente el juego completo en el que el usuario puede seleccionar el objeto elegido y la máquina juegue utilizando el algoritmo anteriormente implementado.
-

IMPORTANTE: La parte de prácticas de la asignatura consistirá en la resolución e implementación de la relación de prácticas del curso 2012/2013. Se deberá entregar la memoria y el código de las prácticas en un CD el día del examen. Dicha memoria contendrá los siguientes apartados para cada una de las prácticas:

- Análisis del problema resuelto
- Explicación del esquema algorítmico utilizado
- Pseudocódigo de algoritmo principal
- Resultados obtenidos y conclusiones

Además se realizará una prueba en el laboratorio que consistirá en una modificación de una de las prácticas de la relación.