Nombre y Apellido: ...................................................................................N° Legajo: .................

#### Recuperatorio Segundo Parcial de Estructuras de Datos y Algoritmos

#### Segundo Cuatrimestre de 2016

|  |  |  |  |
| --- | --- | --- | --- |
| Ejercicio 1 | ***Ejercicio 2*** | ***Ejercicio 3*** | Nota |
|  |  |  |  |

Condición Mínima de Aprobación: Tener por lo menos dos ejercicios con B-

## Ejercicio 1

La NASA ha decidido finalmente enviar hombres a Marte. Se realizaron análisis sobre qué lugares son los más interesantes para visitar y le piden a Ud. un experto en algoritmos que dado el mapeo del planeta y los lugares interesantes que éste cuenta, decida en dónde conviene aterrizar la nave a fin de minimizar la distáncia de la nave a todos los puntos interesantes. Hay que tener en cuenta que el terreno de Marte es irregular y hay algunos lugares intransitables por tierra.

Se pide: dada la matriz que representa el mapa de Marte implementar un algoritmo que devuelva las coordenadas óptimas para el aterrizaje a fin de minimizar la distáncia a todos los lugares interesantes.

public class MarsMap {

private enum TerrainType { NORMAL, INTERESTING, INTRANSITABLE };

private TerrainType[][] map;

public Point bestLandingPlace() {

...

}

}

Ejemplo:

Dada esta matriz se obtendría el punto (2, 2) dado que la distáncia a cada punto interesante minimal (4, 4, 3 y 3)

(para simplificar notamos con una X los lugares intransitables, con una I los interesantes, con una A la respuesta y vacío los normales)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I |  |  |  |  |  |
| X | X |  |  |  |  |
| X | X | A | X | I |  |
| X |  |  |  | I |  |
| X | I |  | X | X |  |
|  |  |  |  |  |  |

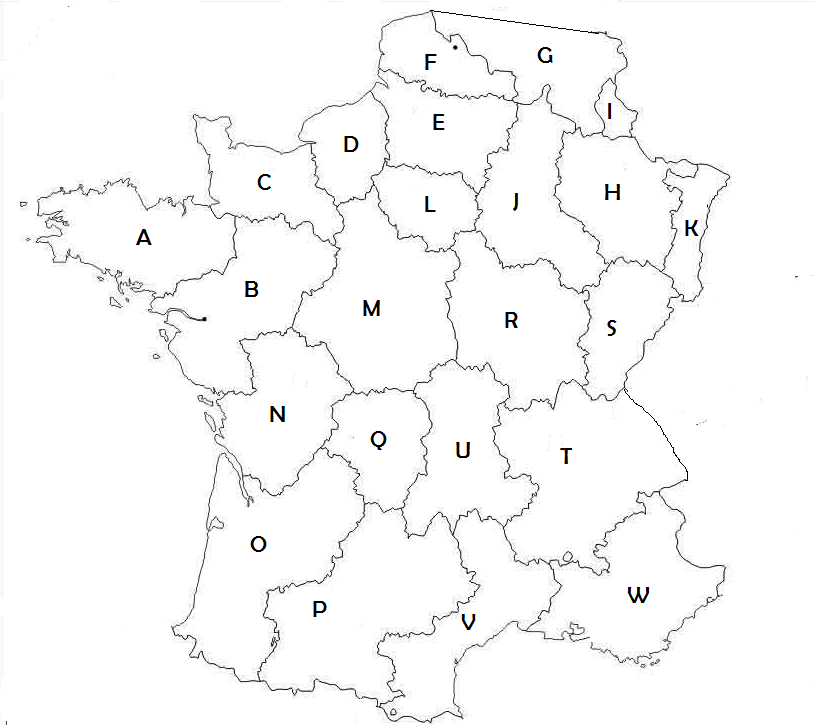
## Ejercicio 2

En una celda de una prisión se encuentra una persona que se desea liberar. No es posible penetrar por debajo del piso (haciendo un túnel) ni por el techo, por lo que la única forma es penetrar primero en una celda que da al muro exterior y luego ir pasando por diversas celdas hasta llegar a la celda destino.

Se conoce el mapa del presidio y también el costo de pasar de una celda a otra.

Se pide determinar por qué celdas se debe ir para liberarlo con el menor costo posible.

Ejemplo: En el siguiente mapa las celdas A, C, D, E, F, G, I, H, K, S, T, W, V, P, O, N y B están conectadas directamente con el exterior. Y se sabe que el prisionero está en alguna de las celdas de A a la W. El costo de pasar de una celda X a otra celda Y es el mismo que pasar de la celda Y a la celda X.



-HAY TAMBIEN UN COSTO PARA PENETRAR A LAS CELDAS EXTERIORES

Cada celda se identifica con un string.

Escribir una clase que soporte la información del mapa y que ofrezca un método que dado una celda donde está el prisionero retorne cuál es el camino con menor costo para liberar al prisionero.

## Ejercicio 3

Basándonos en la clase GraphAdjList vista en clase implementar un algoritmo que dado un peso y un valor de origen devuelva todos los caminos que tengan como origen el nodo con dicho valor y cuyo peso sea menor al valor del peso

public Set<List<V>> avaliablePaths(V origin, int weight)

Ejemplo

avaliablePaths(“A”, 5) en este grafo:

../../../../Downloads/Untitled%20Diagram%20(1).pn

Devolvería:

{[A], [A,B], [A,C], [A,C,D]}