

## Epreuve 2 : Trajet le plus court

Vous devez maintenant proposer un algorithme qui trouve le chemin le plus court entre le départ et l'arrivée .... le minotaure n'étant pas endurant, ce sera plus efficace.

**Coup de pouce :** Vous devez modifier le tableau *laby* de manière à faire apparaître le plus court chemin vers la sortie pour ensuite indiquer la trajectoire au minotaure. Pour ce faire, on partira de la sortie (qui sera notée 1) et on appellera récursivement votre fonction sur chacune des cases adjacentes atteignables (i.e.  $\neq -1$ ) dont la distance à la sortie est moins bonne que la distance calculée.

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	99	99	99	99	99	-1	99	-1
-1	99	-1	-1	-1	99	-1	99	-1
-1	99	-1	99	99	99	-1	99	-1
-1	99	99	99	99	-1	-1	99	-1
-1	-1	-1	-1	99	99	-1	99	-1
-1	99	99	-1	99	-1	-1	99	-1
-1	99	99	99	99	99	99	99	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

*laby* avant l'appel de la fonction

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	13	14	13	12	11	-1	7	-1
-1	12	-1	-1	-1	10	-1	6	-1
-1	11	-1	9	8	9	-1	5	-1
-1	10	9	8	7	-1	-1	4	-1
-1	-1	-1	-1	6	7	-1	3	-1
-1	8	7	-1	5	-1	-1	2	-1
-1	7	6	5	4	3	2	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

*laby* après l'appel de la fonction

Il suffit ensuite de suivre le chemin en allant dans la case avec le nombre immédiatement inférieur. Si le minotaure est posé en [1][1] orienté vers le bas l'ordre à lui passer est : TTTGTTDTTGTT

**Vous devez déposer et tagger sur le gitlab :**

- une documentation expliquant votre algorithme
- le code python de votre implémentation
- faites une animation graphique de la progression de l'algo en utilisant pygame, par exemple
- Générer les ordres au format TTTGTTDTT....