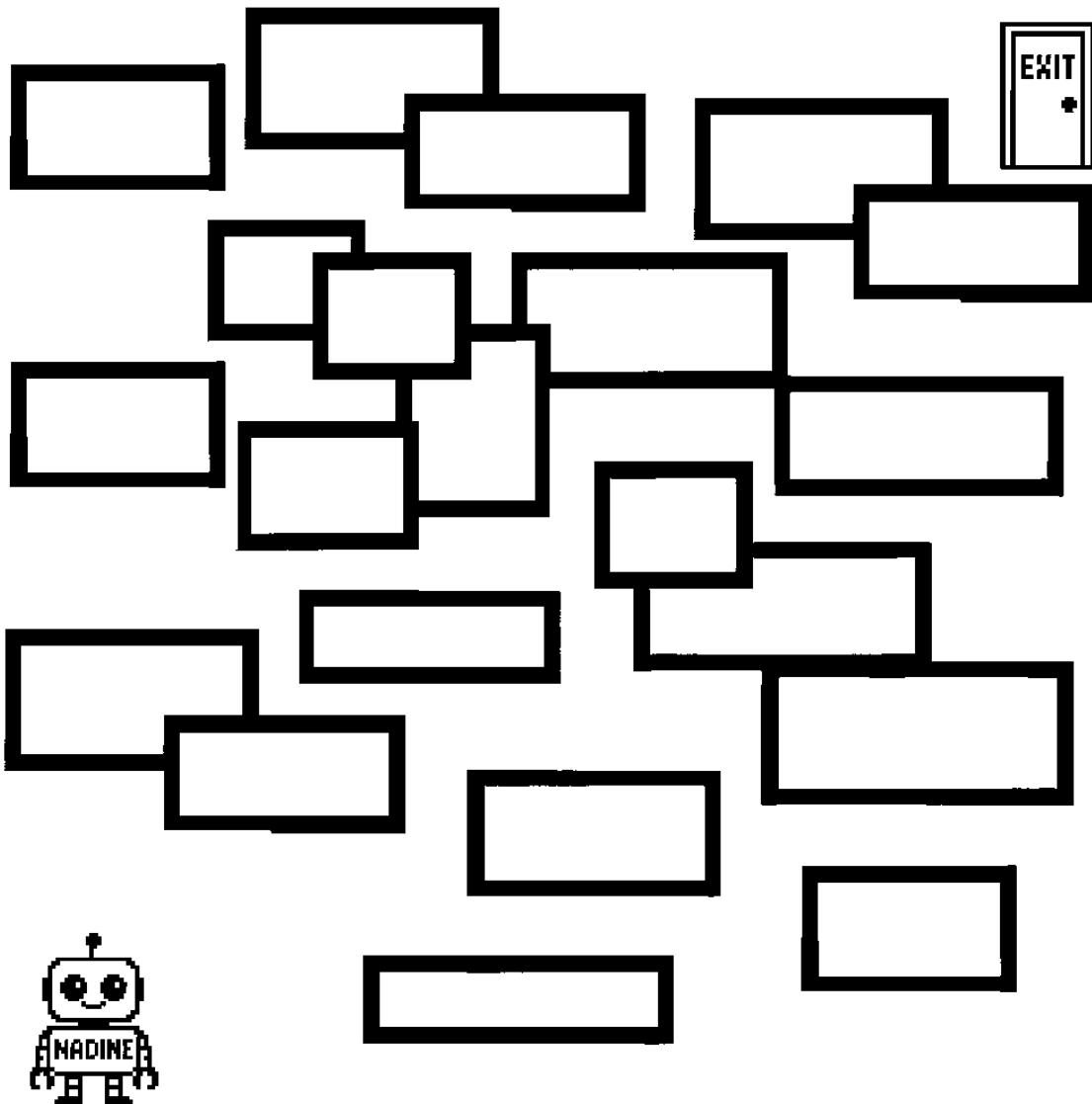


Chemin le plus cours



chemin le plus cours

Le but du TP est de trouver le chemin le plus court dans une image noir et blanc.

I Construction du graphe

On peut obtenir la matrice noire et blanche d'une image par le code suivant

```
from PIL import Image
import numpy as np

img = Image.open("obsacles.png", "r")
matriceRGB = np.array(img).tolist()
img_nb = img.convert("1")
matriceNB = np.array(img_nb).tolist()
```

```
    img.close()

    #Conversion en Image
    Image.fromarray(np.array(matriceRGB,dtype=np.uint8), mode="RGB").show()
```

Exercice 1. *Graphe d'image*

Implémentez une fonction `grapheIMG(m) -> dict` qui prend en entrée une matrice représentant une image en noir et blanc et qui renvoie le dictionnaire d'adjacence du graphe $G = (S, A)$ tel que S est l'ensemble des couples (i, j) des coordonnées des pixels blancs de l'image et chaque arête de A relie deux pixels blancs adjacents, i.e à 1 pixels de décalage en abscisse ou en ordonnée.

Exercice 2. *Algorithme de recherche de plus court chemin.*

Pour trouver un chemin optimal on peut faire une parcours en largeur du graphe en gardant un historique h . h est un dictionnaire tel que pour tout sommet s visité, $h[s]$ est le sommet depuis lequel on a ajouté s .

Écrire une fonction qui prend un graphe en dictionnaire d'adjacence et renvoie un historique du parcours.

Exercice 3. *Coloration du chemin*

Grâce à l'historique on peut colorier le chemin dans l'image.

- Calculer le graphe de l'image « obstacles.png »
- Récupérer l'historique du parcours depuis le pixel (900, 200) jusqu'au pixel (70, 1000).
- En partant de (70, 1000) et tant que l'on a pas atteint un sommet s qui n'a pas d'historique, on remonte l'historique et on sauvegarde chaque pixel rencontré dans une liste.
- On reprend la matrice couleur de « obstacles.png » et on colorie les pixels sauvegardé en rouge.

Exercice 4. *Recherche du chemin avec un parcours en profondeur*

Essayer d'adapter un parcours en profondeur pour trouver un chemin plus rapidement mais pas forcément optimal.