

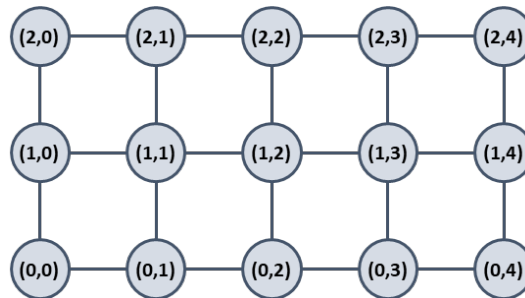
TP 13

Quelques algorithmes de tris

Exercice 1 – Génération et parcours de labyrinthe

Génération d'un labyrinthe

Soit une grille rectangulaire $n \times p$ constituée de n lignes et de p colonnes contenant toutes les arêtes possibles. On modélise cette grille par un graphe dont l'ensemble des sommets est donné par les couples (i, j) tels que $i \in \llbracket 0, n \rrbracket$ et $j \in \llbracket 0, p \rrbracket$. Les voisins d'un sommet (i, j) sont ceux situés en haut, en bas, à droite et à gauche s'ils existent (par exemple, le sommet $(0, 0)$ a comme voisins les sommets $(0, 1)$ et $(1, 0)$).



Le graphe est implémenté par un dictionnaire d'adjacence où les clés sont les tuples, coordonnées d'un sommet. La valeur associée est une liste des sommets voisins.

Ainsi, la grille 2×2 sera modélisée par le graphe suivant

```
{(0,0):[(0,1),(1,0)], (0,1):[(0,0),(1,1)], (1,0):[(0,0),(1,1)], (1,1):[(0,1),(1,0)]}
```

Question 1 Écrire la fonction `creer_graphe(n:int, p:int) -> {}` permettant de créer une grille de n lignes et p colonnes.

Question 2 En utilisant la fonction `tracer_graphe(G)`, vérifier que votre graphe a la bonne structure.