

## 1. Présentation du problème

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P


## 2. Modélisation par un graphe

Proposer deux «formes» possibles pour ce graphe.

 **Correction**

### 3. Implémentation du graphe en Python

En utilisant une classe `Graphe`, implémenter le graphe de ce labyrinthe.

 **Correction**

### 4. Recherche du plus court chemin

En utilisant la fonction `recherche_chemin` du cours, établir le plus court chemin pour aller de A vers P dans ce labyrinthe.

 **Correction**

### 5. Conclusion

Sur un labyrinthe un peu plus imposant, voici l'illustration de notre méthode de résolution :

- le parcours en largeur part découvrir les cases dans toutes les directions.
- lorsque la case cherchée (ici, la rouge) est trouvée, on remonte à chaque case précédente grâce au dictionnaire `parent`, et ainsi le chemin de sortie du labyrinthe est généré.

[Code de cette animation](#) (en Pygame)