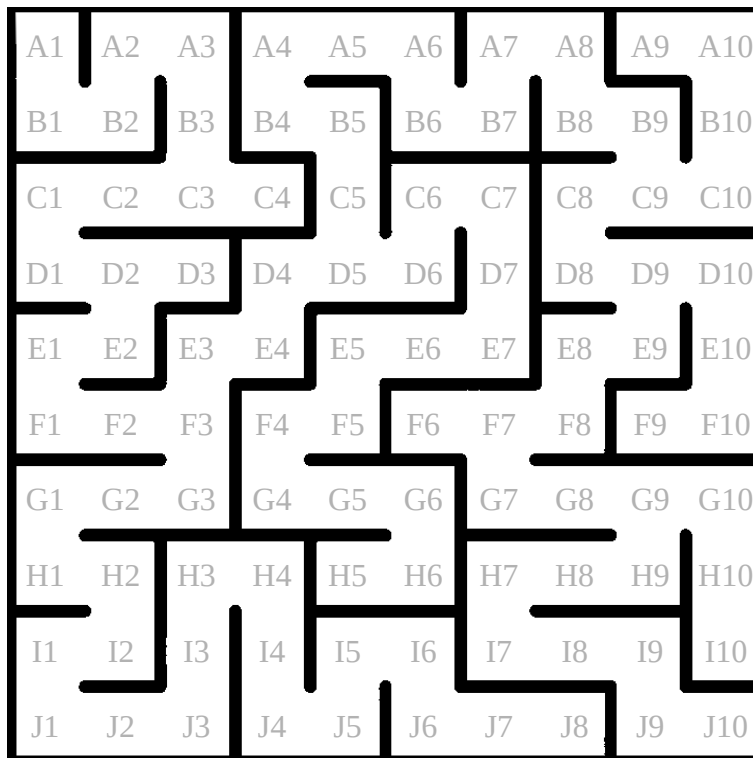


# TD Labyrinthe



**Liste d'adjacence du graphe associé au labyrinthe :**

{ 'A1' : ['B1'], 'A2' : ['A3'],

**Utilisation de la classe Graphe pour implémenter le labyrinthe :**

# création d'une instance de graphe non orienté nommé « labyrinthe »

# ajout d'un arc entre les sommets de clés 'A1' et 'A2' :

```
# labyrinthe.dict_adj = { 'A1' : ['B1'], 'B1' : ['A1'] }
```

# ajout d'un arc entre les sommets de clés 'A2' et 'A3' :

```
# labyrinthe.dict_adj = { 'A1' : ['B1'], 'A2' : ['A3'],  
                          'A3' : ['A2'], 'B1' : ['A1'] }
```

...

**Il ne reste plus qu'à utiliser les méthodes de recherche de chemins :**

# recherche des chemins entre les sommets de clés 'A1' et 'J10'

# recherche des chemins entre les sommets de clés 'A6' et 'I10'

**Et pour le chemin le plus court ?**