

# Comparer des algorithmes pour des problèmes classiques

## Tâches à réaliser :

1. Modéliser le plateau comme une grille d'objets noeuds en mémoire
2. Initialiser les liaisons sous la forme d'un arbre **quaternaire** d'objets (Noeud) Python, reliés les uns aux autres.

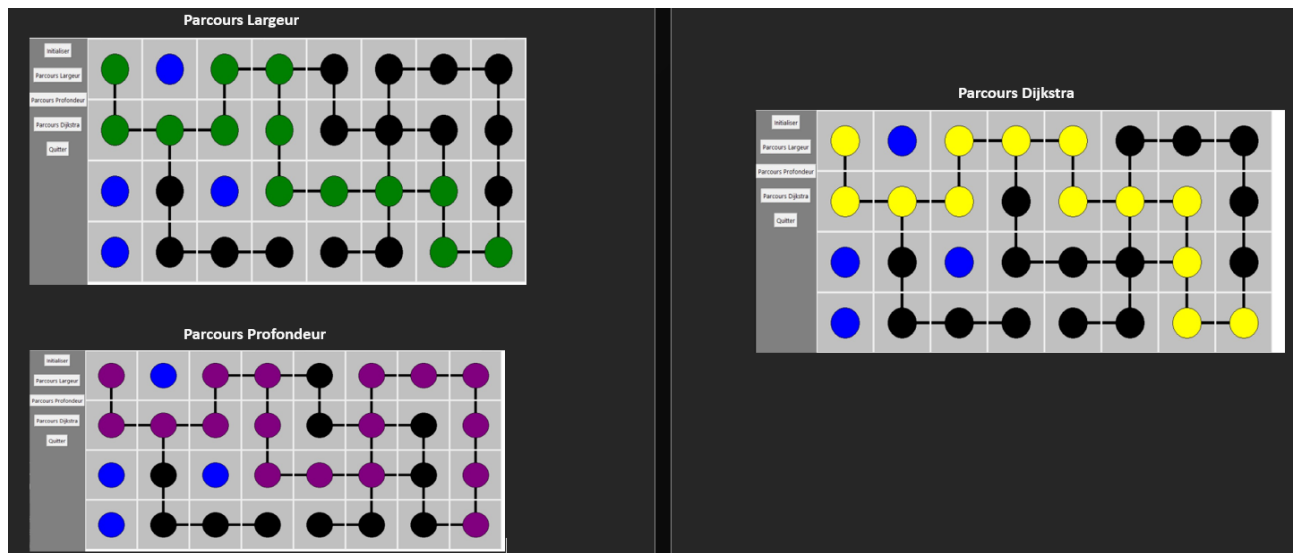
Chaque Noeud aura pour propriétés :

- nom : string # coordonnées (colonne, ligne)
- poids : entier # uniquement pour Dijkstra
- nord, sud, est, ouest : Noeud # noeuds adjacents
- statut : string # "aucun" / "visité" / "actif"

3. Implémenter la recherche de la sortie avec chacun des algorithmes suivants :

- parcours en profondeur
- parcours en largeur
- algorithme de Dijkstra

Le parcours du graphe sera journalisé dans un fichier.



Je suis en capacité de comparer des algorithmes pour des problèmes classiques. Dans la SAÉ 2.02 « Exploitation algorithmique d'un problème » j'ai analysé les performances du parcours en profondeur, en largeur et de l'algorithme de Dijkstra pour résoudre un labyrinthe, en mettant en évidence leurs avantages et limites selon le contexte d'utilisation.