



Travail pratique #5

Travail présenté à

Ettore Merlo

De la part de

Miah, Mohammad Jamil - 2184862

Rouleau, Thomas - 2221053

Dans le cadre du cours de

INF2010 - Structures de données et algorithmes

26 novembre 2023

Polytechnique Montréal

Pour analyser et comparer les performances des algorithmes BFS et DFS que nous avons implémentés pour résoudre le labyrinthe, on peut se baser sur plusieurs critères, notamment l'utilisation de la mémoire et le nombre de tuiles visitées.

Nos résultats

BFS:

```
✓ Tests passed: 11 of 11 tests - 292 ms
Nodes in Stack current iteration: 13
Nodes in Stack current iteration: 13
Nodes in Stack current iteration: 14
Total Nodes Traversed: 10998
Max Stack Size: 30
BUILD SUCCESSFUL in 2s
4 actionable tasks: 1 executed, 3 up-to-date
4:16:06 p.m.: Execution finished ':test --tests "BFSMazeTest"'.

```

DFS:

```
✗ Tests failed: 8, passed: 3 of 11 tests - 285 ms
Nodes in Stack current iteration: 50
Nodes in Stack current iteration: 55
Nodes in Stack current iteration: 55
Total Nodes Traversed: 9524
Max Stack Size: 95

Expected :684
Actual   :706

```

Allocation de Mémoire

BFS:

Il utilise une Queue pour stocker les nœuds à visiter. Dans notre cas, le "Max Stack Size" est de 30. Cela signifie qu'à un moment donné, le BFS a eu jusqu'à 30 nœuds en mémoire.

DFS:

Il utilise une pile pour stocker les nœuds à visiter. Le "Max Stack Size" pour DFS est de 95, ce qui est significativement plus élevé que celui de BFS.

En conclusion, en moyenne, BFS alloue moins de mémoire que DFS, car il maintient moins de nœuds en mémoire à un moment donné.

Visite de Tiles

BFS:

Il visite les nœuds niveau par niveau. Le "Total Nodes Traversed" est de 10998, indiquant le nombre total de tuiles visitées.

DFS:

Il visite les nœuds en explorant aussi profondément que possible dans chaque branche avant de revenir en arrière. Le "Total Nodes Traversed" est de 9524.

En conclusion, sur les labyrinthes testés, DFS a visité moins de tuiles en moyenne pour trouver la sortie. Cela pourrait s'expliquer par le fait que DFS peut trouver une route plus directe vers la sortie en allant en profondeur, tandis que BFS explore plus largement.

Donc, en gros, BFS est généralement plus efficace pour trouver le chemin le plus court dans un labyrinthe, car il explore tous les chemins possibles niveau par niveau. Cependant, cela peut entraîner une exploration plus large, donc potentiellement plus de tuiles visitées. D'autre part, DFS peut être moins efficace pour trouver le chemin le plus court, car il explore en profondeur, mais cela peut se traduire par une visite de moins de tuiles si le chemin est trouvé rapidement. Cependant, le DFS peut se retrouver bloqué dans des impasses ou explorer des chemins inutilement longs.