

INF2010

Structures de données et algorithmes

*Laboratoire 5 – Graphes*

Soumis par :

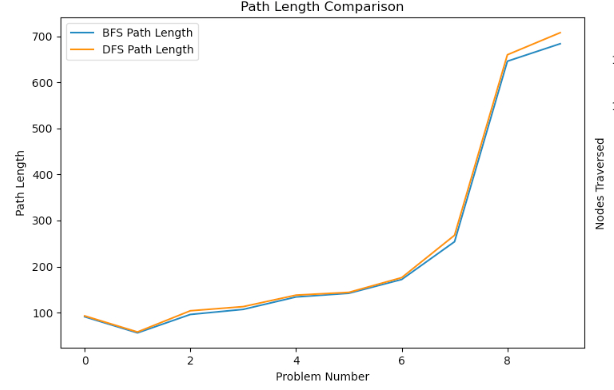
*Kandarakis Desrosiers, Charles – 2179269*

*Yassine Seddaoui - 2204548*

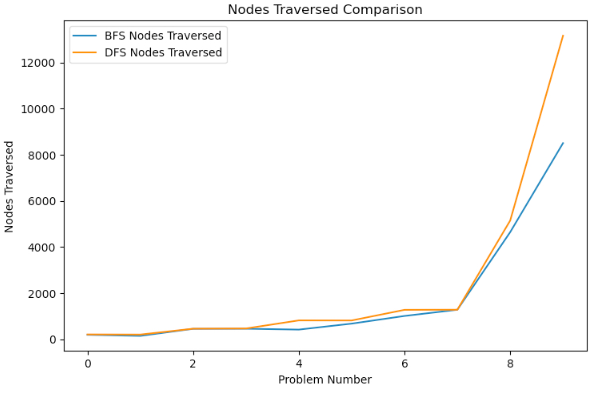
(groupe 3)

*Le 26 novembre 2023*

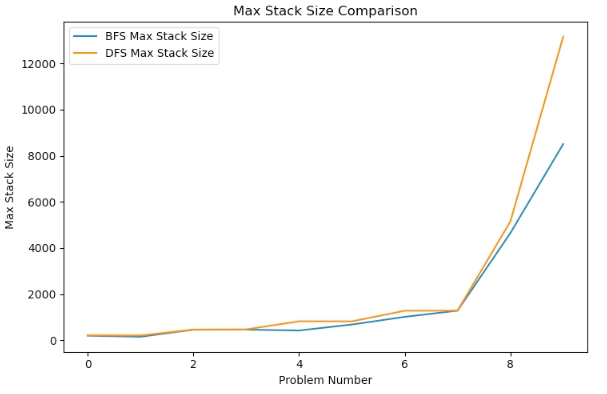
Analyse visuelle des algorithmes (BFS, DFS) pour la résolution d’un labyrinthe :



*Figure 1*



*Figure 2*



*Figure 3*

Réponses aux sous-questions:

Quel algorithme alloue le moins de mémoire en moyenne:

À l’aide de la figure 3, on peut constater que l’algorithme BFS alloue le moins de mémoire en moyenne. Normalement, BFS utilise plus de mémoire que DFS. Cependant, comme BFS utilise généralement O(maxWidth) mémoire et DFS utilise O(maxDepth) mémoire il est possible que la mémoire utilisée par BFS soit plus petite si maxWidth < maxDepth. Cela est surement dû au fait que notre labyrinthe ait plusieurs lignes droites.

Quel algorithme alloue le plus de mémoire en moyenne:

À l’aide de la figure 3, nous pouvons voir que l’allocation de mémoire semble égale pour les deux algorithmes pour un petit nombre de nœuds. Par contre, nous pouvons analyser une divergence entre les deux algorithmes plus les nombres de données est important. Cette divergence s’explique par le fonctionnement de ces deux algorithmes. L’algorithme DFS garde dans la queue le chemin direct de la racine (début du labyrithe) à la feuille (la fin du chemin). Pour ce qui est de l’algorithme BFS, toutes les feuilles sont gardées dans la queue. Du a la conception des labyrinthes (nous estimons qu’il contient plusieurs lignes droites, donc moins de birfucation), l’algorithme DFS alloue plus de mémoire que BFS.

Quel algorithme visite le plus de Tile en moyenne:

À l’aide de la figure 2, on peut constater que l’algorithme DFS visite le plus de “Tile” en moyenne. DFS va visiter des nœuds et continuer en profondeur jusqu’à ce qu’il atteigne la sortie. Par la suite, il revient sur ses pas pour visiter un autre chemin. Ainsi, il va visiter tous les nœuds. BFS va plutôt visiter les nœuds par niveau et s’arrêter immédiatement après avoir trouvé la sortie.

Quel algorithme visite le moins de Tile en moyenne:

Pour répondre à cette question, analysons le graphique en figure 2. L’axe X défini les fichiers d’entrées de type CSV. L’axe Y, quant à elle défini le nombre de nœuds visité par l’algorithme utilisé. Nous pouvons voir qu’à partir du fichier csv 04, une petite divergence est distinguable entre les deux algorithmes. Ainsi, à partir du fichier 07, on voit clairement que l’algorithme DFS visite de plus en plus de nœuds. De ce fait, nous pouvons conclure que l’algorithme BFS est plus performant et semble plus complet.