

Exo 1

1.1 ① cf code python

② pour BFS et DFS, b = le facteur de branchement vaut : 3

BFS soit $b=3$
 d = profondeur de la solution dans l'arbre

complexité \leadsto temps : $O(3^d)$
espace : $O(3^d)$

DFS soit $b=3$
 m = profondeur maximum d'une feuille
($m \geq d$)

complexité \leadsto temps : $O(3^m)$
espace : $O(3 \cdot m)$

③ problèmes DFS:

Solution {
• La solution n'est pas forcément optimale
• Dans cet exercice on ne voit pas ce problème, mais si on avait un facteur de branchement plus grand, il serait possible de se "perdre" dans l'arbre en explorant le côté opposé à la solution
→ recherche en profondeur limitée

problème BFS:

Solution {
• On perd beaucoup de temps avec des stratégies qui ne mènent pas à la solution, vu que c'est une exploration de toutes les possibilités.
• On doit aussi stocker en mémoire beaucoup plus de nœuds visités
→ recherche bidirectionnelle
→ recherche en coût uniforme

Bilan: DFS est un pari "tout ou rien", alors que BFS, ou pari sur toutes les possibilités,

→ le meilleur cas en DFS est beaucoup meilleur que le meilleur cas en BFS

Solution | on peut améliorer les deux avec l'introduction d'une heuristique qui nous permettra de faire un pari raisonné, mais ça c'est le prochain chapitre.

④ pour savoir laquelle des deux techniques appliquer nous devons nous poser quelques questions:

- c'est quoi le facteur de branchement?
- $m \gg d$ vs $m = d$?
- y-a-t'il plusieurs façons d'atteindre notre solution?

entre autres

Dans notre cas:

	noeuds explorés	longueur solution	noeuds stockés en mémoire	Score
BFS	24	8 ✓	24 + 2	1
DFS	9 ✓	10	9 + 5 ✓	2

nombre itérations

→ représentatif du temps

nombre états dans la solution y compris initial et final

états déjà visités + états à visiter

?
à l'arrêt de l'algorithme

BFS \leadsto solution optimale

DFS \leadsto usage de ressources moins important

réponse : Vu que le facteur de branchement est petit pour ce problème, je suis de l'avis que DFS est plus adopté, sauf si on cherche la solution optimale à tout prix.

exo 2

$$1 = O(100)$$

$$1 < O(n)$$

$$n = O(10n + 5)$$

$$n^2 > O(100n)$$

$$n < O(n^2)$$

$$10n^3 + n^2 - 5n + 100 = O(n^3)$$

$$n^{50000} + 1000000 < O(n^2)$$

$$n 2^n > O(2^n)$$