

# TP3

---

Auteurs : Karam ELNASORY · Erkin Tunc BOYA · Semih DOYNUK

---

Exercice 1

Exercice 2

Exercice 3

---

## Exercice 1

---

### Question 1

Un graphe d'état peut contenir des cycles alors que l'arbre de recherche ne peut pas.

### Question 2

Un graphe d'état ne mène pas toujours à un arbre de recherche, ce premier doit être connexe, c'est à dire que quelque soient x et y sommets du graphe, il existe un chemin entre x et y.

### Question 3

Un état est une description de la situation actuelle alors que le noeud est une solution partielle au problème, on peut d'ailleurs avoir plusieurs noeuds différents qui font référence au meme états.

## Exercice 2

---

### Question 1 :

```
public class ResearchAlgorithm {

    public static Node search(KnightsTourProblem problem)
    {
        int counter = 0;
        List<Node> frontier = new LinkedList<>();

        Node root = new Node(problem.initialState(), null, null);
        frontier.add(root); // Starts with the root node

        while ( ! frontier.isEmpty() )
        {
            Node currentNode = frontier.remove(0);
            counter += 1;

            if (problem.isGoalState(currentNode.getState())) // Condition d'arrêt qu'on va exploiter pour faire un tour fermé
```

```

        {
            System.out.println("Found a solution after evaluating " + counter + " nodes.");
            return currentNode;
        }

        frontier.addAll(currentNode.expand(problem)); // Exploration des enfants du noeud courant ⇒
        parcours en largeur
    }
    return null;
}
}

```

C'est une parcours en profondeur en construisant le graph de tous les états possible (brute force) jusqu'à arriver à un état solution.

→ justification : regardez les commentaire sur le code.

---

### Question 2 :

Le cavalier commence sur une case donnée, ne visite chaque case qu'une seule fois, et doit parcourir toutes les cases, et à chaque fois qu'il visite une case 8 mouvements sont possible au plus.

⇒ n étant le nombre de case ( $n = m*m$ ),  $O(n) = n^8$  noeuds à stocker .

Pour une grille 5\*5 :  $n = 25$  ,  $O(25) = 25^8 = 152,587,890,625$

Pour une grille 6\*6 :  $n = 36$  ,  $O(25) = 36^8 = 2,821,109,907,456$

On observe un comportement exponentielle par rapport au nombre de cases.

---

### Question 3 :

Code sur le git [/TP3/Ex2\\_Tour\\_Cavalier\\_DFS](#)

---

### Question 4

---

### Question 5

---

## Exercice 2

### Question 1

Le DFS itératif sera surement le + efficace (pas encore implémenté)

### Question 2

Il faudrait garder les états parcouru dans une liste et a chaque état crée on vérifie s'il existe déjà dans la liste des états parcouru ou alors dans la frontière.