

TP numéro 1

Intelligence artificielle, ENSIIE

Semestre 4, 2021–22

Le but de ce TP est d'implémenter le jeu du taquin et les différents algorithmes de recherche. Votre code devra être déposé à la fin du TP sur exam.ensiie.fr dans le dépôt `ia-fisa-tp1`.

Exercice 1 : Interface de jeu

1. Définir le type des configurations.
2. Implémenter une fonction d'affichage d'une configuration donnée.
3. Définir un type pour les opérateurs (déplacer le trou vers le haut, la droite, le bas, la gauche).
4. Écrire une fonction qui prend une configuration et un opérateur et qui retourne la configuration atteinte, ou une erreur si le mouvement n'est pas possible.
5. Écrire une fonction qui teste si une configuration est finale.
6. Écrire une fonction qui retourne une configuration aléatoire. (Idée : partir de la configuration finale, puis déplacer le blanc 10000 fois de façon aléatoire.)

Exercice 2 : Recherche aveugle

1. Implémenter, ou utiliser une bibliothèque existante, des files contenant des configurations.
2. Implémenter la recherche en largeur d'abord.
3. Implémenter, ou utiliser une bibliothèque existante, des piles contenant des configurations.
4. Implémenter la recherche en profondeur d'abord.
5. Implémenter la recherche en profondeur limitée.
6. Implémenter la recherche en profondeur itérative.
7. Instrumenter les fonctions pour récupérer le nombre de nœuds traités et le nombre de coups à jouer depuis l'état initial.
On pourra également instrumenter le code pour afficher la profondeur quand on est en train de traiter un nœud à une profondeur pas encore atteinte jusque là.
8. Tester à partir d'une configuration aléatoire.

Exercice 3 : Algorithme A*

1. Implémenter les trois heuristiques vues en cours.
2. Implémenter, ou utiliser une bibliothèque existante, des files de priorité contenant des configurations.
3. Implémenter l'algorithme A* en prenant une heuristique en paramètre.
4. Instrumenter les fonctions pour récupérer le nombre de nœuds traités et le nombre de coups à jouer depuis l'état initial.
On pourra également instrumenter le code pour afficher la profondeur et l'heuristique quand on est en train de traiter un nœud à une profondeur pas encore atteinte jusque là ou à une heuristique plus faible que celles vues jusqu'à présent.
5. Tester à partir d'une configuration aléatoire.