$h(u) \geq ?$
$h(n) = 0$

Un noeud qui apparait 2 fois dans la file est normal :)

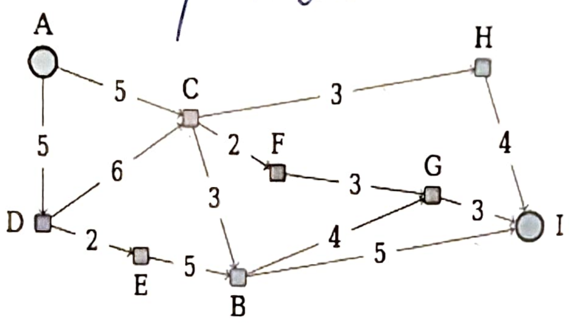
Intelligence Artificielle TD3

Résolution des problèmes - Algorithmes de recherche

Exercice 1

Considérez la carte suivante. Le but est de trouver le chemin ~~le plus court~~ de A vers I.

Dans quel ordre les noeuds sont visités ?



Donner l'ordre de parcours des noeuds pour les algorithmes :

1. largeur d'abord
2. profondeur d'abord

Une heuristique h est donnée comme suit : Le plus CH.

Noeud	A	B	C	D	E	F	G	H	I
h	10	4	5	10	10	3	3	4	0

Y a-t-il un chemin à partir de A ≤ 10 ? Non

— Est-ce que h est admissible ?

— Appliquez la recherche gloutonne en utilisant h.

— Appliquez la recherche A^* en utilisant h.

Exercice 2

Nous considérons un monde avec 4 pions (A,B,C,D) non superposables. Ils peuvent être arrangés dans n'importe quel ordre, sauf que A qui ne peut pas être plus à droite que D. Par exemple, ABCD et CBAD sont deux états possibles du monde, tandis que DCBA et CDAB ne sont pas possibles. Le monde peut être manipulé par une action de la forme echange(x, y) qui échange les pions des positions x et y. Par exemple echange(1, 2) transforme BCAD dans CBAD. Seules les actions echange(1, 2), echange(2, 3) et echange(2, 4) sont autorisées. Ils donnent un successeur uniquement si la situation atteinte est possible.

1. Dessinez le graphe d'états.
2. On suppose que l'état de départ est ADBC et l'état que l'on veut atteindre est CBAD. On suppose que chaque action coûte 1. Donnez une "bonne" heuristique h admissible (une heuristique qui ne surestime pas le coût) pour ce problème. Remarquez que chaque action déplace 2 pions.
3. Appliquez la recherche gloutonne avec votre heuristique. Ne considérez pas les noeuds déjà développés. En cas d'égalité choisissez un noeud à développer au hasard.
4. Appliquez la recherche A* avec votre heuristique. Ne considérez pas les noeuds déjà développés. En cas d'égalité choisissez un noeud à développer au hasard.

Exercice 3

Le taquin est un jeu solitaire en forme de damier. Il est composé de 15 petits carreaux numérotés de 1 à 15 qui glissent dans un cadre prévu pour 16. Il consiste à remettre dans l'ordre les 15 carreaux à partir d'une configuration initiale quelconque.

Dans la figure 1 nous illustrons une version du jeu avec 8 cases. A gauche une situation de départ quelconque, et à droite la situation but désirée.

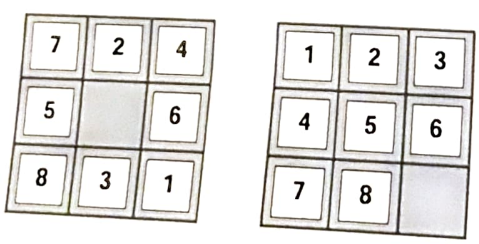


FIGURE 1 – Exemple du jeu du taquin en version 8.

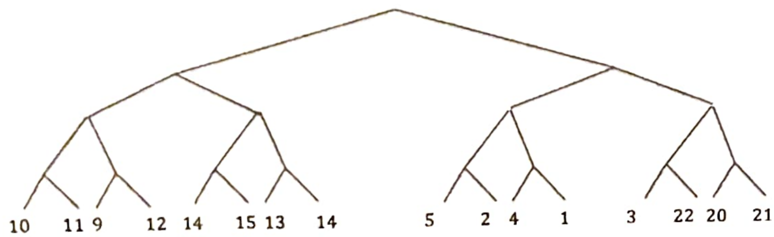
1. Proposez une modélisation des état du jeu

états finaux (les noeuds feuilles), vous utiliserez les valeurs suivantes :
1 si le **joueur 1** gagne, -1 s'il perd.

3. Qui est le vainqueur ?

Exercice 6

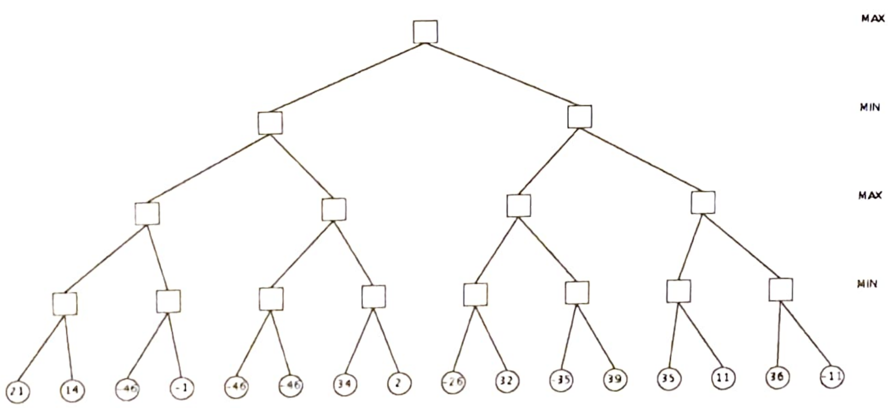
Soit l'arbre de jeu suivant.



1. Appliquez l'algorithme Alpha-Beta sur cet arbre

Exercice 7

Appliquez l'algorithme Alpha-Beta sur l'arbre de jeu suivant :



2. En utilisant cette modélisation, écrivez les états de départ et de fin illustrés sur la figure 1.
3. Quel sont les actions possible dans ce jeux ?
4. Quel est le facteur de branchement moyen dans l'arbre du jeu ? (pour cela identifiez le nombre d'actions possibles dans les différentes positions possibles pour le carré vide.)
5. Nous proposons de résoudre le ce problème en utilisant une stratégie A*. Pour cela, nous utiliserons la fonction heuristique suivante : ■ h : le nombre de pièces mal placées ■.
   — Développez l'arbre du problème à la profondeur 2, en indiquant sur chaque noeud les valeurs de h et de g (rappelez vous que g est le coût du chemin parcouru, ici c'est le nombre de mouvements effectués).
6. Proposez une autre fonction heuristique admissible, et refaites la question précédente.

Exercice ~~5~~6 :

Soit le jeux suivant à deux joueurs. Initialement, on dispose d'une pile de n jetons. A tour de rôle, les joueurs doivent diviser une des piles devant eux en deux piles **non vides** et de **tailles différentes**. A chaque tour du jeu, on crée donc une nouvelle pile.

Le joueur qui ne peut plus jouer (i.e. il ne reste que des piles de 1 ou 2 jetons) a perdu. Voici un exemple d'une partie du jeu :

- **Initialement** : une pile avec 6 jetons. La configuration initiale = (6).
- **Joueur 1 :** divise la pile de 6 en deux piles de 4 et de 2 jetons. La configuration résultante = $(4, 2)$.
  PS : une autre solution pour le joueur 1 aurait été de diviser la pile en 5 et 1 jetons = $(5, 1)$.
- **Joueur 2 :** divise la pile de 4 jetons en deux piles de 3 et de 1 jetons. La configuration résultante = $(3, 1, 2)$
- **Joueur 1 :** divise la pile de 3 jetons en deux piles de 2 et de 1 jetons. La configuration résultante =$(2, 1, 1, 2)$
- **Joueur 2 :** a perdu

1. *Construire l'arbre complet du jeu en partant d'une pile de 7 jetons.*
2. *Supposons que le* **joueur 1** commence. Appliquez l'algorithme Min-*Max sur l'arbre du jeu construit précédemment.* Pour les valeurs des

IA : TD°3

...icit :

...ut est de trouver : ~~un chemin de A vers I~~

...cours largeur : dans quel ordre les noeuds sont visités ?

Noeuds ouverts visités	File
$\emptyset$	A
A	C - D
A - C	D - H - F - B
A - C - D	H - F - B - EC
A - C - D - H	F - B - E - C - I
A - C - D - H - F	B - E - C - I - F
A - C - D - H - F - B	E - C - I - F
	E - C - I - G - G - I

ACDHFBECI FGIBHFB

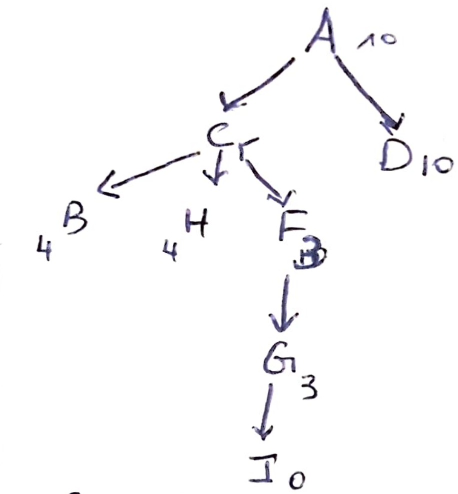
Parcours profondeur

Noeuds visités	Pile
$\emptyset$	A
A	D, C
A, D	C, E, C
A, D, C	B, F, H, E, C
A, D, C, B	I, G, F, H, E, C
A, D, B, B, I	G, F, H, E, C

* Admissible ?

1) $\forall n \; h(n) \geq 0$
2) $h(n) = 0$ si n : état final
3) $h(n) = \infty$ si état final non atteignable à partir de n
4) h ne surestime pas le coût réel vers l'état final. c'est nécessaire pour que A* retourne la solution optimale

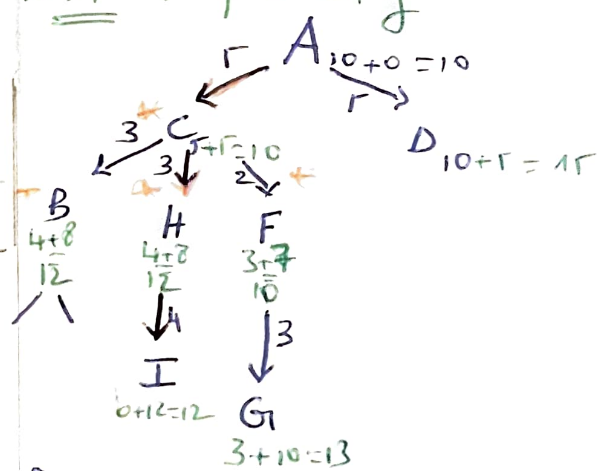
* Gloutonne ?



$A \xrightarrow{5} C \xrightarrow{2} F \xrightarrow{3} G \xrightarrow{3} I$

le coût = 13.

* A* ? $f = h + g$



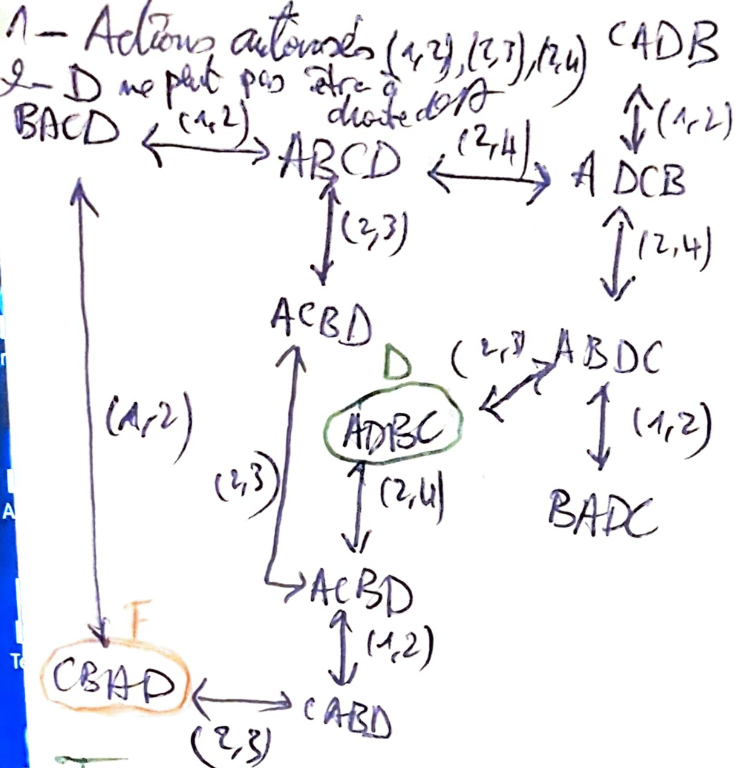
P.C.C : $A \longrightarrow C \longrightarrow H \longrightarrow I$

le coût = 12.

Exercice 2 :

1 — Actions autorisées (1,2), (2,3), (2,4)

2 — D ne peut pas être à droite de A



Trouver une heuristique h admissible ?

h_1 " distance de chaque pion par rapport à sa position dans l'état final "

h_1 n'est pas admissible car elle surestime le coût vers l'état final.

Par exemple :

$h_1(ABDC) = 2 + 0 + 1 + 3 = 6 > 4$ échanges.

Cependant il ~~faut~~ suffit de 4