

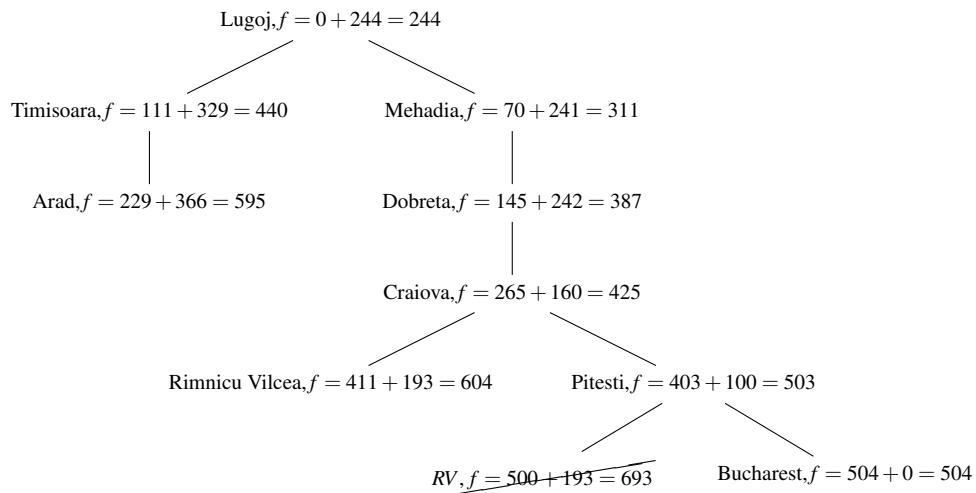
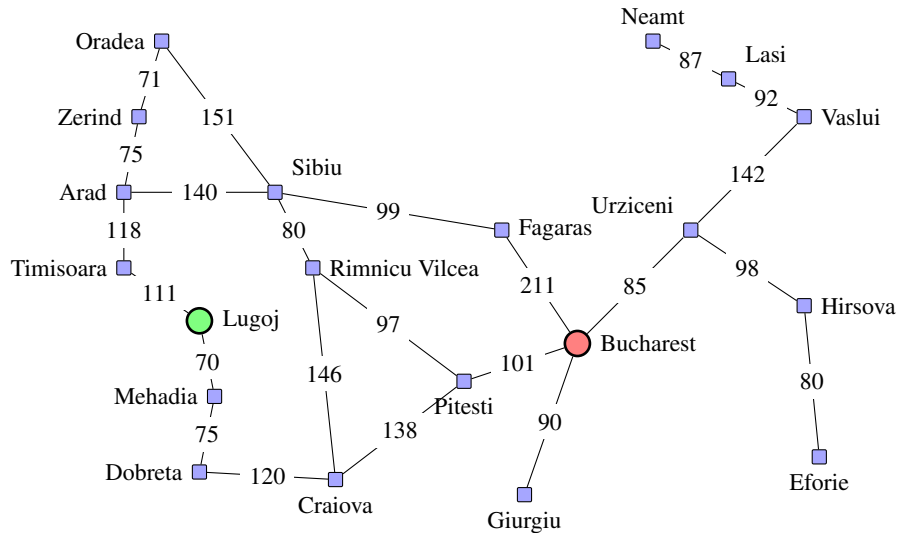
Intelligence Artificielle – TD 3

ALGORITHMES ET RECHERCHES HEURISTIQUES

CORRECTION

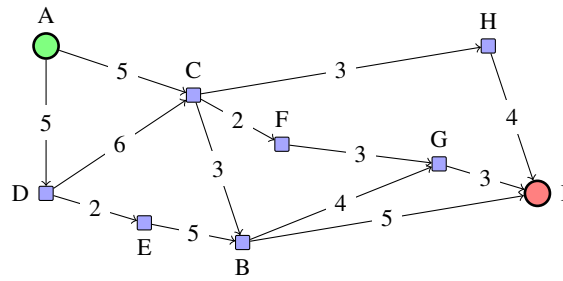
Exercice 1 - Appliquez l'algorithme A* au problème du voyage en Roumanie en appliquant l'heuristique de la distance à vol d'oiseau. Vous supposerez que vous voulez voyager de Lugoj à Bucharest. Pour chaque nœud, vous donnerez les valeurs de f , g et h . Si un même état apparaît dans deux nœuds différents, avec deux valeurs de f différentes, on conserve seulement celui avec la meilleure (la plus petite) valeur de f . On supposera aussi que l'on ne passera pas deux fois par la même ville sur le même chemin (la même branche de l'arbre de recherche).

Ligne droite jusqu'à Bucharest					
Arad	366	Hirsova	151	Rimnicu Vilcea	193
Bucharest	0	Lasi	226	Sibiu	253
Craiova	160	Lugoj	244	Timisoara	329
Dobreta	242	Mehadia	241	Urziceni	80
Eforie	161	Neamt	234	Vaslui	199
Fagaras	176	Oradea	380	Zerind	374
Giurgiu	77	Pitesti	100		



Liste des noeuds développés : Lugoj, Mehadia, Dobreta, Craiova, Timisoara, Pitesti, Bucharest

Exercice 2 - Considérez la carte orientée suivante. L'objectif est de trouver le chemin le plus court de A vers I. On donne également trois heuristiques, h_1 , h_2 et h_3 .



Nœud	A	B	C	D	E	F	G	H	I
h_1	10	5	5	10	10	3	3	3	0
h_2	10	2	8	11	6	2	1	5	0
h_3	10	2	6	11	9	6	3	4	0

1. Est-ce que h_1 , h_2 et h_3 sont admissibles ? Justifier.

- h_2 n'est pas admissible : $h_2(C) > h^*(C)$
- h_1 et h_3 sont admissibles : $\forall n, h_1(n) \leq h^*(n)$ et $h_3(n) \leq h^*(n)$

2. Quelles relations de dominance existent entre ces trois heuristiques ?

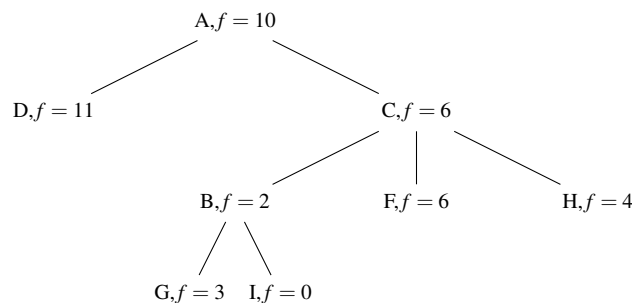
- h_2 n'est pas admissible, donc ne peut pas dominer/être dominée par une autre heuristique.
- h_1 ne domine pas h_3 : $h_3(F) > h_1(F)$
- h_3 ne domine pas h_1 : $h_1(B) > h_3(B)$

3. Est-ce que $h_4 = \max(h_1, h_3)$ est admissible ? Justifier.

Nœud	A	B	C	D	E	F	G	H	I
h_4	10	5	6	11	10	6	3	4	0

h_4 est bien admissible : comme $\forall n, h_1(n) \leq h^*(n)$ et $h_3(n) \leq h^*(n)$, alors $\max(h_1(n), h_3(n)) \leq h^*(n)$

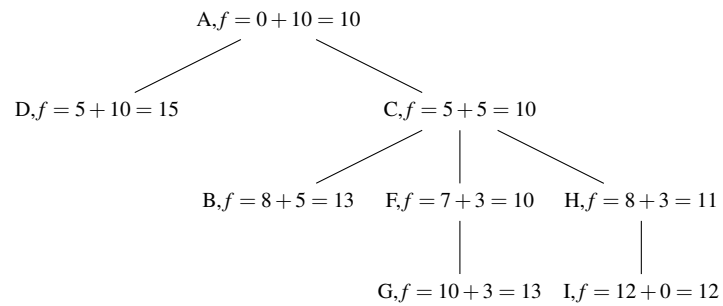
4. Appliquer la recherche gloutonne en utilisant h_3 . Donner la suite des nœuds développés.



Liste des nœuds développés : A, C, B, I

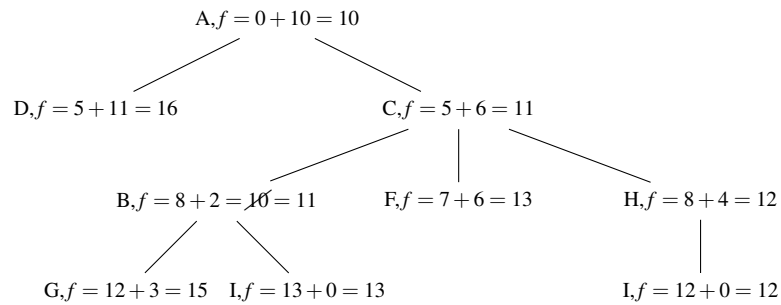
5. Appliquer la recherche A* en utilisant h_1 , puis h_3 , puis h_4 . Donner à chaque fois la suite des nœuds développés.

A* avec h_1 .



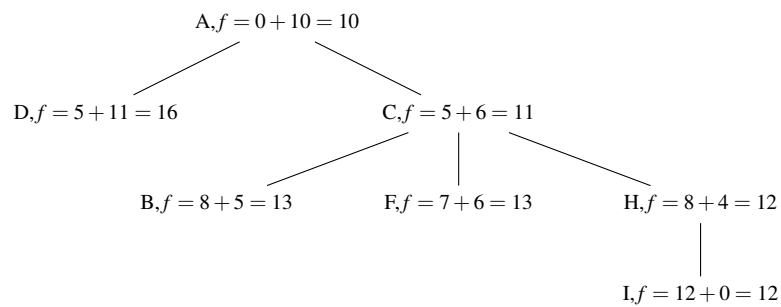
Liste des nœuds développés : A, C, F, H, I

A* avec h_3 . **Attention, la valeur de f ne doit pas décroître!**



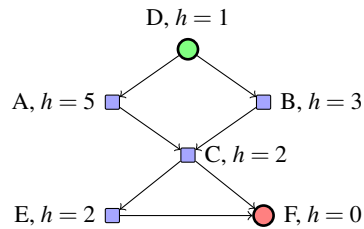
Liste des nœuds développés : A, C, B, H, I

A* avec h_4 .



Liste des nœuds développés : A, C, H, I

Exercice 3 - Considérez l'espace de recherche suivant (D est l'état initial, F est l'état final) :

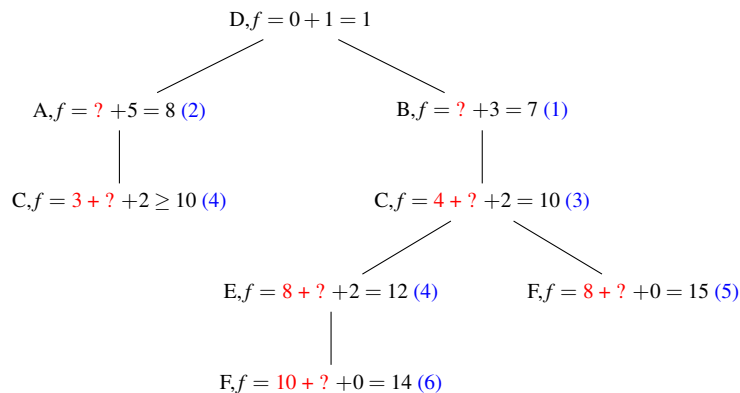


Pour chaque nœud est indiquée la valeur de l'heuristique h . On veut récupérer le coût de chaque arc entre deux nœuds. Pour cela nous disposons d'une trace de l'algorithme A^* . Pour chaque pas de l'algorithme est indiquée la liste des nœuds encore à traiter avec la valeur $f = g + h$. Si un nœud peut apparaître deux fois avec deux valeurs de f différentes, on conserve seulement celui avec la meilleure (la plus petite) valeur de f .

[(D, $f = 1$)]
 [(B, $f = 7$) , (A, $f = 8$)]
 [(A, $f = 8$) , (C, $f = 10$)]
 [(C, $f = 10$)]
 [(E, $f = 12$) , (F, $f = 15$)]
 [(F, $f = 14$)]

Utiliser cette trace et votre connaissance du fonctionnement de A^* pour calculer les coûts de tous les arcs.

Pour trouver les coûts des arcs, on reconstruit l'arbre obtenu en appliquant A^*



Détail du raisonnement :

1. On développe D. On en déduit :

- (1) le chemin [D,B] coûte 4
- (2) le chemin [D,A] coûte 3

2. On développe B. On en déduit :

- (3) le chemin [B,C] coûte 4

3. On développe A. On en déduit :

- (4) on a gardé qu'un nœud C dans la trace de l'algorithme, avec un $f(C) = 10$. Le nœud trouvé en développant A n'est donc pas conservé, et on sait que la valeur $f(C)$ sur ce chemin est ≥ 10 . On peut donc seulement en déduire que le chemin [A,C] a un coût ≥ 5

4. On développe C. On en déduit :

- (5) le chemin [C,E] coûte 2
- (6) le chemin [C,F] coûte 7

5. On développe E. On en déduit :

- (7) le chemin [E,F] coûte 4

Exercice 4 - Soient les grilles de taquin suivantes :

Etat initial

1	2	3
4	6	8
7	5	

Etat final

1	2	3
4	5	6
7	8	

Nous considérons l'heuristique suivante :

- $h(n)$ = la distance de Manhattan totale (la distance de chaque pièce entre sa place actuelle et sa position finale en nombre de places)

Appliquez la recherche A* en utilisant h . Donner la suite des nœuds développés. Si un même état apparaît dans deux nœuds différents, avec deux valeurs de f différentes, on conserve seulement celui avec la meilleure (la plus petite) valeur de f . On supposera aussi que l'on ne passera pas deux fois par la même état sur le même chemin (la même branche de l'arbre de recherche).

