UNIVERSITE CADDI AYYAD
FACULTE DES SCIENCES SEMLALIA
INFORMATIQUE LPI

Exercice 1

1) La complexité en temps de l'algorithme de recherche *en largeur d'abord* pour un facteur de branchement *b* et une profondeur *p* est en :

a- $O(b^p)$
b- $O(b+p)$
c- $O(p^b)$
d- $O(b*p)$

2) L'algorithme de recherche en profondeur est un algorithme :

a- adéquat
b- complet
c- optimal

SOLUTION

1)

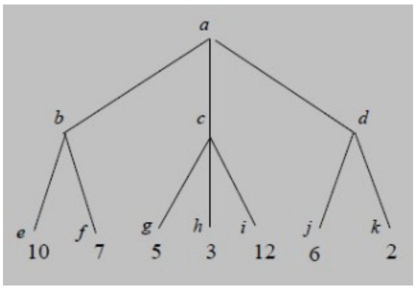
a) $O(b^p)$
~~b) $O(b+p)$~~
~~c) $O(p^b)$~~
~~d) $O(b*p)$~~

2) Adéquat

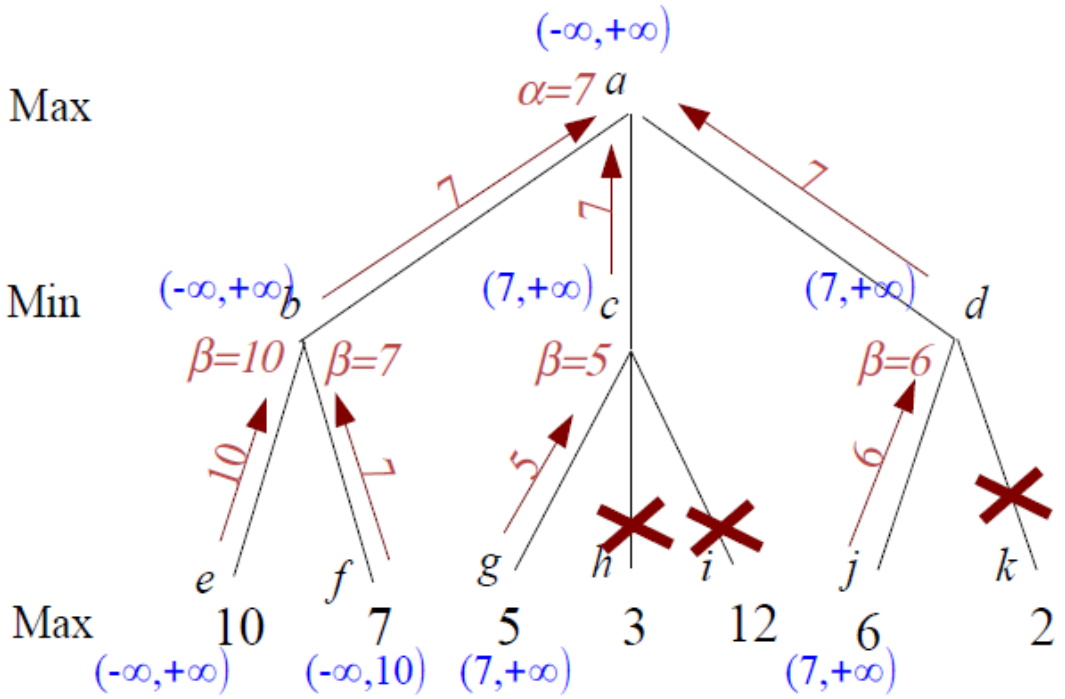
Exercice 2 :

Soit l'arbre de jeu ci dessous, la racine *a* étant un noeud *Max*.

Question : Exécutez l'algorithme de gauche à droite sur cet arbre de jeu. Vous prendrez soin d'expliquer chaque étape importante.



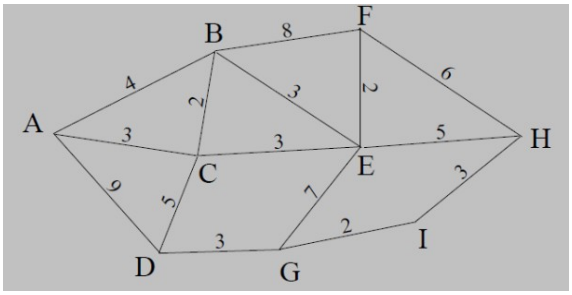
Solution :
En bleu les valeurs d'appel de ㄱㅅㄷet ㄴㅈ, en rouge les valeurs de retour.



Explication :
L'algorithme calcule sur chaque noeud une valeur de retour. Sur les noeuds Max, on affecte au maximum entre sa valeur courante et la valeur de retours de chacun des fils des noeuds. De même sur un noeud Min, on affecte au minimum entre la valeur courante et la valeur de retour de chacun des fils. Si après avoir exploré tous les fils
d'un noeud max (resp. min) ⌜α⌝ < ⌞β⌟ on renvoie ⌞β⌟ (resp. ⌜α⌝)
Dès que ⌜α⌝ devient supérieur ⌞β⌟ (i.e. sans calculer la valeur de retour des autres fils du noeud courant) :
– sur un noeud max : on retourne ⌞β⌟,
– sur un noeud min : on retourne ⌜α⌝

Exercice 3 :

Soit le graphe suivant, la valeur portée sur chaque arc correspond au coût de passage d'une extrémité de l'arc à l'autre. On souhaite calculer le plus court chemin de A à I.



On a de plus la fonction heuristique *h* qui estime le coût pour atteindre I depuis chaque sommet. *h* est donnée par le tableau ci dessous.

	A	B	C	D	E	F	G	H	I
h	12	7	10	4	5	6	2	2	0

Question : Appliquez l'algorithme A* avec la fonction *h* sur ce graphe.

SOLUTION

On représente le déroulement de l'algorithme sous forme de tableau. Chaque ligne du tableau correspond à un tour de boucle de l'algorithme. La première colonne indique le numéro de tour de l'algo, dans la deuxième colonne on indique le sommet choisi. Dans la troisième colonne l'ensemble des ouverts de l'étape courante est représenté, chaque état est noté sous le format « sommet(g(x),f(x)=g(x)+h(x)) ». La dernière colonne contient l'ensemble des fermés de l'étape courante.

Etape	***Choix***	***Ouverts***	***Fermés***
Init		{A(0,12)}	{}
1	A(0,12)	{B(4,11);C(3,13);D(9,13)}	{A(0,12)}
2	B(4,11)	{C(3,13); D(9,13);E(7,12);F(12,18)}	{A(0,12); B(4,11)}
3	E(7,12)	{C(3,13); D(9,13);**F(9,15)**;G(14,16);H(12,14)}	{A(0,12); B(4,11); E(7,12)}
4	C(3,13)	{**D(8,12)**;F(9,15);G(14,16);H(12,14);**E(6,11)**}	{A(0,12); B(4,11); C(3,13)}
5	E(6,11)	{D(8,12);**F(8,14)**;**G(13,15)**;**H(11,13)**}	{A(0,12); B(4,11); C(3,13); E(6,11)}
6	D(8,12)	{F(8,14);**G(11,13)**;H(11,13)}	{A(0,12); B(4,11); C(3,13); E(6,11); D(8,12)}
7	G(11,13)	{F(8,14);H(11,13);I(13,13)}	{A(0,12); B(4,11); C(3,13); E(6,11); D(8,12); G(11,13)}
8	I(13,13)		

Remarques :
– A l'initialisation, on met dans Ouverts, le sommet de départ;
– A chaque étape on choisit dans Ouverts un sommet s tel que f(s)=g(s)+h(s) soit minimal. Pour tous les voisins v de s, si v n'appartient ni à Ouverts ni à fermés, on ajoute v à Ouverts. Sinon on remet v dans Ouverts avec une nouvelle valeur de g(v) seulement si g(s)+coût(s->v) est inférieur à la valeur de g(v) mémorisée.