

- Appliquez l'algorithme $\alpha\beta$ sur ce même arbre.
Vous redévelopperez complètement sur une feuille séparée l'arbre exploré. Indiquez pour chaque coupe effectuée, s'il s'agit d'une coupe α ou β et s'il s'agit d'une coupe de surface ou d'une coupe profonde.
- Proposez un ordre de développement des fils de chaque noeud permettant de minimiser le nombre de noeuds développés par $\alpha\beta$.

Exercice 3 Iterative Deepening $\alpha\beta$

On considère l'algorithme $\alpha\beta$, dans sa version où il exécute une recherche à profondeur incrémentale. On suppose que lors de la première exploration de chaque nouveau niveau de l'arbre, les différents fils d'un même nœud sont explorés suivant l'ordre lexicographique.

- Détailler le fonctionnement de cet algorithme sur le graphe décrit implicitement dans le tableau suivant :

noeud n	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
$h(n)$	13	8	15	12	7	55	19	28	17	11	14	15	10	11	
$succ(n)$	B C D	E F G	H I J	K L	M N	O P	Q R S	T U	V W	X Y	Z A1	A2 A3	a b	c d	
noeud n	O	P	Q	R	S	T	U	V	W	X	Y	Z	A1	A2	A3
$h(n)$	20	13	10	11	6	14	30	18	19	12	10	14	12	15	8
$succ(n)$	e	f	g	h i	j k	l	m n	o	p	q r s	t u	v w	x1 x2	y1 y2	z1 z 2
noeud n	a	b	c	d	e	f	g	h	i	j	k	l	m	n	
$h(n)$	30	12	8	15	19	3	7	17	12	10	6	14	12	8	
noeud n	o	p	q	r	s	t	u	v	w	x1	x2	y1	y2	z1	z2
$h(n)$	10	8	12	8	15	6	7	14	19	6	14	15	16	12	10

- Comment gérer en pratique le réordonnement des fils tout en limitant la consommation mémoire ?

Exercice 4 Etude analytique de l'algorithme Iterative Deepening $\alpha\beta$

On suppose qu'une machine met T_h secondes pour évaluer la valeur heuristique d'une configuration de jeu.

- Expliquez pourquoi on peut, dans les questions suivantes, négliger le temps de création des noeuds internes de l'arbre de recherche pour estimer le coût total des différents algorithmes.
- Donnez le temps nécessaire à MiniMax pour estimer la valeur de la racine d'un arbre de jeu dont le facteur de branchement est b et l'horizon est fixé à p (en demi-coups).
- Donnez un encadrement du temps nécessaire à $\alpha\beta$ pour effectuer la même estimation.
- Imaginons que l'on utilise un algorithme de type MiniMax à profondeur incrémentale, oubliant tout entre deux recherches successives. En toute généralité, quel serait le pourcentage de feuilles supplémentaires développées par un tel algorithme, par rapport à un MiniMax classique, pour arriver à la même profondeur, en fonction de b et de p ?
- On souhaite à présent comparer la version classique de l'algorithme $\alpha\beta$ avec sa version à profondeur incrémentale. Supposons qu'à profondeur égale, une version à profondeur incrémentale explore 50% de nœuds en plus que la taille de l'arbre critique (optimal) et qu'une version $\alpha\beta$ coupe la moitié des noeuds explorés par MiniMax (c'est une supposition réaliste : cela simule le fait que la version iterative deepening va utiliser les lancements précédents pour ordonner les fils et se rapprocher plus de l'arbre critique, alors que la version directe n'a pas de moyen simple de se rapprocher de l'arbre critique). Comparez les deux approches en termes de noeuds supplémentaires développés.

Vous prendrez, pour les applications numériques, $T_h = 1/1000000s$, $b = 30$ et $p = 10$.

Exercice 5 Jeux avec tirage de dés

On désire adapter le principe de la recherche $\alpha\beta$ au cas des jeux à deux joueurs dont les mouvements possibles dépendent d'un tirage de dés, effectué avant que chaque joueur ne décide quoi jouer (par exemple le jeu des petits chevaux).

1. Adaptez l'algorithme MiniMax pour ce problème.
2. Comment transformer votre algorithme pour avoir des solutions optimistes ou pessimistes par rapport aux tirages possibles de dés ?
3. Est-il encore possible de trouver des stratégies gagnantes ?
4. Peut-on (sous quelles conditions) adapter l'algorithme $\alpha\beta$ pour ce problème ?