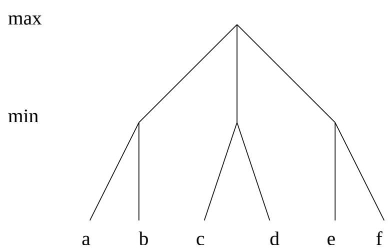
Université Paris 7 - Master 1 Informatique - Intelligence Artificielle
Partiel du 9 novembre 2005 - Durée : 1 heure et demie

Informations : Tous les documents reliés sont autorisés. Le barème est donné à titre indicatif et peut être modifié.

Exercice 1 Algorithme α-β (3 points)
Considérez l'arbre de jeux suivant:

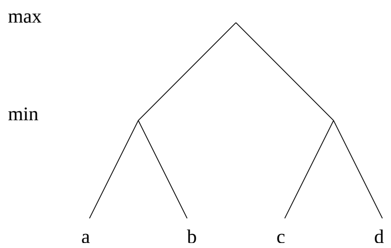


La racine est un nœud max.

- Donnez des valeurs (toutes **différentes** les unes aux autres) aux feuilles de sorte que l'algorithme α-β coupe d'une part
  - au moins une feuille avec un parcours de gauche à droite, et d'autre part
  - au moins une feuille avec un parcours de droite à gauche

  Attention: Les valeurs doivent être les mêmes pour les deux parcours.

Exercice 2 Algorithme α-β (3 points)
Considérez l'arbre de jeux suivant:



La racine est un nœud max. Les valeurs a,b,c et d sont toutes **différentes** les unes aux autres.

- Montrez qu'on ne peut pas donner des valeurs a,b,c,d aux feuilles telles que l'algorithme α-β coupe des feuilles pour **chacun** des deux parcours:
  - de gauche à droite
  - de droite à gauche

  Indication: Donnez les conditions pour qu'une coupure ait lieu pour les deux parcours et montrez une contradiction.

Exercice 3 Jeux (4 points)
On considère le jeu à deux joueurs suivant: On commence avec une pile de 7 jetons. Pendant le jeu, plusieurs piles vont être crées. Chaque joueur doit diviser une pile en deux piles non vides et de **tailles différentes** (par exemple s'il y a trois piles à 2, 3 et 2 jetons, le seul coup possible est de diviser la pile de 3 jetons en 2 et 1 jetons). S'il ne peut plus jouer, le joueur a perdu. On appelle les deux joueurs Max et Min. Si joueur Max gagne la valeur de la position est 1. Si joueur Min gagne la valeur de la position est -1.

- Dans ce jeu on sait que les seules valeurs possibles sont -1 et 1. On peut donc commencer l'algorithme α-β avec $\alpha = -1$ et $\beta = 1$. Pourquoi ?
- Appliquez l'algorithme α-β au jeu. Commencez avec $\alpha = -1$ et $\beta = 1$. Max commence.
- Qui gagne le jeux ?

Exercice 4 Classification (3 points)
Le classificateur naïf de Bayes pour une description $d = (d_1, \ldots, d_n)$ est défini comme suit:

$$C_{NaiveBayes}(d) = \underset{k \in \{1,\ldots,c\}}{argmax}\ (\Pi_{i \in \{1,\ldots,n\}} \widehat{P}(d_i/k)) \times \widehat{P}(k)$$

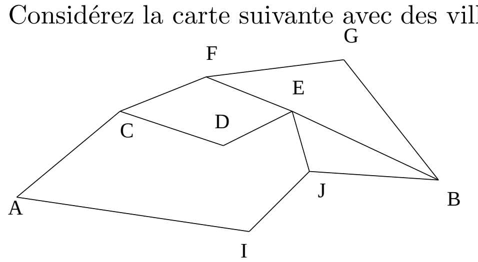
où $\widehat{P}(d_i/k)$ est la proportion d'éléments de classe k qui ont une description d_i dans l'échantillon. et $\widehat{P}(k)$ est la proportion d'élément de classe k dans l'échantillon.

On considère le problème de classification suivant: Il y a deux classes (*anglais*, *francais*) et trois attributs: taille (*petite*,*grande*), age (*jeune*,*vieux*) et revenu (*bas*,*haut*). On a l'échantillon S suivant, donné comme un ensemble de vecteurs $(taille, age, revenu, classe)$: $S = \{(petite, jeune, haut, anglais), (grande, vieux, haut, anglais), (petite, vieux, bas, anglais), (grande, vieux, bas, francais), (grande, jeune, haut, francais), (petite, jeune, bas, francais)\}$.

- Donnez $\widehat{P}(k)$ et $\widehat{P}(d_i/k)$ pour toutes les classes et déscriptions (attributs).
- Donnez le classificateur naïve de Bayes pour les descriptions $(petite, vieux, haut)$ et $(grande, jeune, bas)$.

Exercice 5 (Algorithme A*) (5 points)

Considérez la carte suivante avec des villes:



Les routes entre les villes peuvent contenir des parties montantes, descendantes et plates. Dans les parties montantes on peut avancer à 60 km/h, dans les parties descendantes à 120 km/h et dans les parties plates à 90 km/h. Dans le tableau suivant sont indiqués pour chaque route, le nombre de kilomètres du chemin entre deux villes par partie (montantes, descendantes, plates).

Chemin entre	A et C	A,I	C,D	C,F	D,E	E,J	E,B	F,E	F,G	G,B	I,J	J,B
montante	20	50	10	20	0	0	20	0	50	0	0	10
descendante	20	10	20	0	10	0	30	20	0	60	10	10
plate	15	30	9	21	9	30	0	21	0	0	30	21

On veut trouver le chemin le plus court en **temps** (minutes) entre A et B. Pour cela on veut appliquer l'algorithme A*. On dispose en plus de l'information suivante: Pour chaque ville X on connaît la taille des parties montantes, descendantes et plates sur le chemin à vol d'oiseau entre la ville X et B.

Chemin directe entre	A et B	C,B	D,B	E,B	F,B	G,B	I,B	J,B
montante	40	30	30	40	40	20	20	10
descendante	40	30	20	0	10	30	0	10
plate	30	18	12	0	12	0	12	12

- L'heuristique associant à X le temps de parcours du chemin à vol d'oiseau de X vers B est-elle admissible ?
- Donnez une heuristique h **admissible**. Cette heuristique devrait être **en général** admissible pour des cartes similaires. Expliquez brièvement pourquoi elle est admissible. Donnez pour chaque ville la valeur h avec votre heuristique.
- Appliquez la recherche gloutonne (sans test d'états répétés) avec votre heuristique.
- Appliquez A* (sans test d'états répétés) avec votre heuristique.

Exercice 6 Algorithmes de recherche (4 points)

On suppose qu'on veut chercher un chemin entre A et B dans un graphe. Dans tous les algorithmes de recherche vus en cours, on commence à chercher à partir de A, jusqu'à-ce-qu'on trouve B. On peut aussi commencer avec B et chercher A. On peut aussi chercher en même temps à partir de A vers B et à partir de B vers A. On appelle cette façon de chercher une recherche bidirectionnelle.

- Comment doit-on modifier la recherche en profondeur d'abord pour faire la recherche bidirectionnelle "intelligemment"? Comment doit-on modifier la recherche en largeur d'abord pour faire la recherche bidirectionnelle "intelligemment" ? Quel est le prix à payer ?
- Est-ce que la recherche en largeur d'abord ou la recherche en profondeur d'abord est plus adaptée pour la recherche bidirectionnelle ? Justifiez.