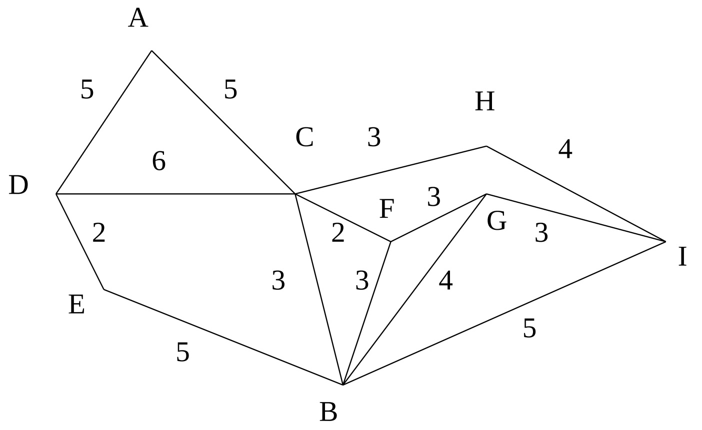
Université Paris 7 - Master 1 Informatique - Intelligence Artificielle
Partiel du 18 novembre 2004 - Durée : 1 heure et demie

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

Exercice 1 Algorithmes de recherche (8 points)
Considérez la carte suivante. Le but est de trouver le chemin le plus court de A vers I.



Le coût de chaque connexion est indiqué. Deux heuristiques h_1 et h_2 sont données comme suit :

Noeud	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	9	6	3	4	0
h^*	12	5	7	12	10	6	3	4	0
h_3	10	5	8	11	10	6	3	4	0

1. Est-ce que h_1 et h_2 sont admissibles ? Justifiez.
   On calcule d'abord h^*, le vrai coût (donné dans le tableau). On vérifie qu'on a toujours $h_1(n) \leq h^*(n)$ pour tout n mais $h_2(C) > h^*(C)$, donc h_1 admissible mais pas h_2

2. Est-ce que h_1 domine h_2 ou h_2 domine h_1 ? Justifiez.
   Ni l'un ni l'autre. D'après le cours, on ne peut parler de domination que si les deux heuristiques sont adminissibles. Ou alors, on a $h_1(B) > h_2(B)$ et $h_2(C) > h_1(C)$.

3. Est-ce que $h_3 = max(h_1, h_2)$ est admissible ?
   Non. $h_3(C) > h^*(C)$

4. Appliquez la recherche gloutonne en utilisant h_2.
   On peut donner un arbre ou la liste des noeuds. Ici:

   ```
   (A,10)
   (C,8)(D,11)
   (B,2)(H,4)(F,6)(A,10)(D,11)
   (I,0)(G,3)(H,4)(F,6)(E,9)(A,10)(D,11)
   ```

   Et on s'arrête avec I et le chemin A,C,B,I

5. Appliquez la recherche A* en utilisant h_1. Donnez la suite des noeuds développés.
   On indique pour chaque noeud sa valeur $f = g + h$:

   ```
   (A,10=0+10)
   (C,10=5+5)(D,15=5+10)
   (F,10=5+2+3)(H,11=5+3+3)(B,13=5+3+5)(D,15)(A,20=5+5+10)(D,21=5+6+10)
   (H,11)(G,13=5+2+3+3)(B,13)(C,14=5+2+2+5)(B,15=5+2+3+5)(D,15)(A,20)(D,21)
   (I,12=5+3+4)(G,13)(B,13)(C,14)(B,15)(D,15)(C,16=5+3+3+5)(A,20)(D,21)
   ```

   Et on s'arrête avec I et le chemin A,C,H,I.

6. Appliquez la recherche A* en utilisant h_2. Donnez la suite des noeuds développés.
On pourrait dire qu'on ne peut pas faire A* puisque h_2 n'est pas admissible (car A* est définie avec des heuristiques adminissible), ou alors on donne:

```
(A,10=0+10)
(C,13=5+8)(D,16=5+11)
(B,10=5+3+2)(H,12=5+3+4),(F,13=5+2+6)(D,16)(A,20=5+5+10)(D,22=5+6+11)
(H,12),(I,13=5+3+5+0)(F,13)(G,15=5+3+4+3)(D,16)(F,17=5+3+3+6)(C,19=5+3+3+8)(A,20)(E,22=5+3+5+9)(D,22)
(I,12=5+3+4),(I,13)(F,13)(G,15)(C,16=5+3+3+5)(D,16)(F,17)(C,19)(A,20)(E,22)(D,22)
```

Et on s'arrête avec I et le chemin A,C,H,I.

7. Appliquez la recherche A* en utilisant h_3. Donnez la suite des noeuds développés.
On pourrait dire qu'on ne peut pas faire A* puisque h_3 n'est pas admissible (car A* est définie avec des heuristiques adminissible), ou alors on donne:

```
(A,10=0+10)
(C,13=5+8)(D,16=5+11)
(H,12=5+3+4)(F,13=5+2+6)(B,13=5+3+5)(D,16)(A,20=5+5+10)(D,21=5+6+10)
(I,12=5+3+4+0)(F,13)(B,13)(D,16)(C,19=5+3+3+8)(A,20)(D,21)
```

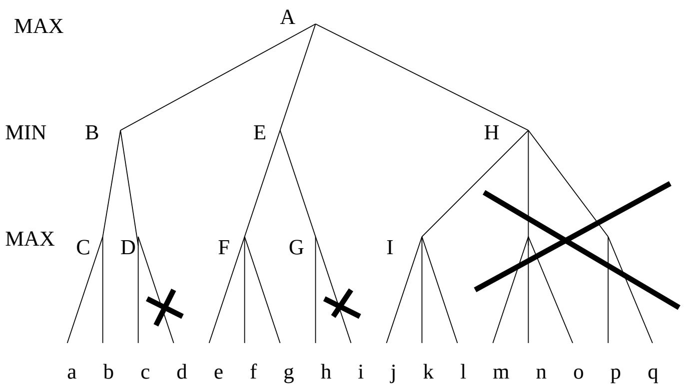
Et on s'arrête avec I et le chemin A,C,H,I.

8. Montrez que pour deux heuristiques admissibles h_1 et h_2, $h_3 = max(h_1, h_2)$ est admissible.
h_1 **et** h_2 **admissible implique pour tout noeud** n $h_1(n) \leq h^*(n)$ **et** $h_2(n) \leq h^*(n)$**. Donc pour tout noeud** n **on a** $max(h_1(n), h_2(n)) \leq h^*(n)$ **donc** $max(h_1, h_2)$ **est admissible.**

9. Si vous avez le choix entre trois heuristiques admissibles h_1, h_2 et $h_3 = max(h_1, h_2)$ laquelle choisissez-vous ? Justifiez brièvement.
h_3 **puisque elle estime le mieux la vraie distance** h^***.**

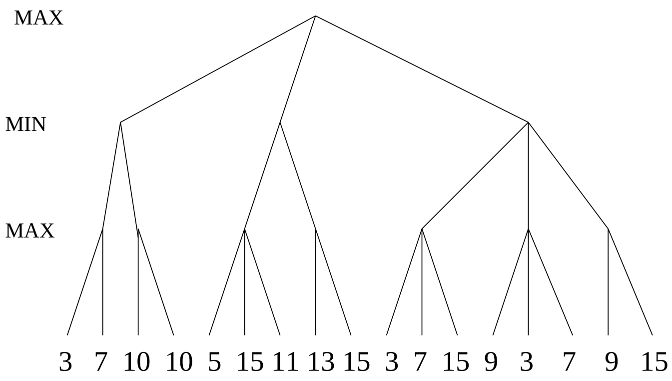
Exercice 2 Jeux (7 points)

1. Considérez l'arbre de jeux suivant :



Donnez des valeurs aux feuilles a jusqu'à q de sorte que l'algorithme α-β coupe exactement les branches indiquées. Appliquez l'algorithme sur l'arbre avec vos valeurs.
Par exemple: a 5, b 2, c 6, e 7, f 1, g 1, h 8, j 1, k 1, l 1
Les autres feuilles peuvent avoir n'importe quelle valeur: Déroulement de l'algorithme: On commence avec A **et** $(\alpha, \beta) = (-\infty, \infty)$**. Ensuite** $B(-\infty, \infty)$**,** $C(-\infty, \infty)$**,**$C(5, \infty)$**,**$B(-\infty, 5)$**,**$D(-\infty, 5)$**,** $D(6, 5)$**, on coupe,** $B(-\infty, 5)$**,** $A(5, \infty)$**,**$E(5, \infty)$**,** $F(5, \infty)$**,** $F(7, \infty)$**,** $E(5, 7)$**,** $G(5, 7)$**,** $G(8, 7)$**, on coupe ,** $E(5, 7)$**,**$A(5, \infty)$**,**$H(5, \infty)$**,**$I(5, \infty)$**,**$H(5, 5)$**,on coupe...**

2. Considérez l'arbre de jeux suivant :



   - Appliquez l'algorithme α-β sur cet arbre en **commençant** avec les valeurs $\alpha = 9$ et $\beta = 14$.
     alpha-beta coupe les deux 10, le 11, et le troisième fils de la racine ! Résultat: 14
   - Appliquez l'algorithme α-β sur cet arbre en **commençant** avec les valeurs $\alpha = 16$ et $\beta = 21$.
     alpha-beta coupe le deuxième fils de chaque noeud min, et le troisième le cas écheant. Résultat: 16
   - Les résultats obtenus ont quelles significations ?
     Prémier résultat: la vrai valeur est ≥ 14**. Deuxième résultat: la vrai valeur est** ≤ 16**.**
   - Sous quelle condition le résultat de l'algorithme α-β avec des valeurs initiales $\alpha = a$ et $\beta = b$ donne le même résultat qu'avec les valeurs initiales $\alpha = -\infty$ et $\beta = \infty$?
     condition: $a \leq$ **le vrai résultat** $\leq b$

3. Pourquoi les algorithmes pour les jeux recherchent toujours à partir de la position courante en avant plutôt que de rechercher en arrière à partir du but ?
   Plusieurs réponses possibles: Il y a plusieurs buts. Comment définir max et min ? , etc.

Exercice 3 Jeux (5 points)

Soit un arbre de jeux **complet** de profondeur p avec facteur de branchement b_1 (chaque noeud a b_1 fils) pour les noeuds MAX et facteur de branchement b_2 (chaque noeuds a b_2 fils) pour les noeuds MIN. La racine est un noeud MAX.

- Donnez le nombre **exactes** de **feuilles** dans l'arbre
  - pour une profondeur p impaire
    $b_1^{(p+1)/2} b_2^{(p-1)/2}$
  - pour une profondeur p pair
    $b_1^{p/2} b_2^{p/2}$
- Donnez le nombre **minimum exacte** de **feuilles** qui sont évaluées (c.-à-d. pour lesquelles la fonction $e(x)$ est appliquée) dans l'algorithme SSS*
  - pour une profondeur p impaire
    $b_1^{p/2} + b_2^{p/2} - 1$: La première passe de l'algorithme va évaluer $b_1^{p/2}$ noeuds. Ensuite, pour confirmer le résultat on doit évaluer encore $b_2^{p/2} - 1$ noeuds.
  - pour une profondeur p pair
    $b_1^{(p+1)/2} + b_2^{(p-1)/2} - 1$

Justifiez vos réponses.