Correction Examen (Session principale)

Filière : GLSI A.U : 2021-2022

Niveau : Deuxième année

Matière : Fondements de l'intelligence artificielle

Exercice1

Question 1 (5 points):

Donner la suite des configurations selon la méthode hill-climbing de **c** parcourues.

★ <u>n=1</u>:

	actuelle, n=1	possibles (c'	est-à-dire niveau suiv	vant),n=2
	c=1	c=2 ?	c=3 ?	c=4 ?
	2 8 3	2 8 3	2 8 3	2 8 3
Configuration	1 6 4	1 4	1 6 4	1 6 4
	7 5	7 6 5	7 5	7 5
f(c)	1+4=5	<u>2+5=7</u> ∨	2+3=5	2+3=5

[→] suite des configurations de c parcourues : 1

★ <u>n=2</u>:

	actu	ielle,	n=2			ро	ss	sibles	(c'es	t-à-dire	: 1	niveau	ı suiv	ant),n	=3			
		c=2			c=5	?			c=6 1	?			c=7 <i>'</i>	?		(c=8 ?	•
Configu ration	1 7	8	3 4 5	2 1 7	8 6	3 4 5		2 1 7	8 4 6	5		2 1 7	8	3 4 5	7		8 1 6	3 4 5
f(c)	2	+5=	7	3	3+4=7			;	3+4=	7		3+	5=8	V		3	3+5=	8

[→] suite des configurations de c parcourues : 1→2

★ <u>n=3</u>:

	ac	tuelle,	n=3			р	ossible	s (c'est-à	n-dire r	niveau su	ivant),	n=4	
		c=7				c=9 '	?			c=10	?		c=11	?
	2		3			2	3		2	8	3	2	3	
Configuration	1	8 4			1	8	4		1		4	1	8	4
	7	8 4 6 5			7	6	5		7	6	5	7	6	5
		•								•			•	
f(c)		3+5=	8		4-	+6=1	0 V			4+5=	:9		4+4=	8

[→] suite des configurations de \mathbf{c} parcourues : $1 \rightarrow 2 \rightarrow 7$

★ n=4:

													
	;	actuelle,	n=4		pos	ssibles	(c'est-à	-dire	ni	veau	suivant),n=5	
		c=9				c=12	?				c=13	?	
Configuration	1 7	2 8 6	3 4 5		7	2 8 6	3 4 5			2 1 7	8	3 4 5	
f(c)		4+6=10			5	+7=12	2 V				5+5=	:10	

 $[\]rightarrow$ suite des configurations de **c** parcourues : $1 \rightarrow 2 \rightarrow 7 \rightarrow 9$

★ <u>n=5</u>:

	ac	tuelle,	n=5		po	ossibles	(c'est-	à-dire	niveau su	ivant),	n=6	
		c=12	2		c=14	?		c=15	?		c=16	?
	1	2	3	1	2	3		2	3	1	2	3
Configuration		8	4	8		4	1	8	4	7	8	4
	7	6	5	7	6	5	7	6	5		6	5
	-				•				•			
f(c)	,	5+7=	12	6	+8=14	V		6+6=	:12		6+6=	12

[→] suite des configurations de **c** parcourues : $1 \rightarrow 2 \rightarrow 7 \rightarrow 9 \rightarrow 12$

★ <u>n=6</u>:

	ac	tuelle,	n=6			pos	sible	s (d	c'est-	-à-dire	n	iveaı	u sui	vant)	,n=7		
		c=14	4		c=17 '	?		С	:=18	?		С	=19	?		c=20	?
Configurati on	1 8 7	6	3 4 5	7	2 8 6	3 4 5	1 8 7		2 4 6	5		1 8 7	2 6	3 4 5	1 8 7	6	3 4 5
f(c)	(3+8=	14	7+7	7=14,	No	7	+7	'=14	, No		7+7	=14	, No	7+	7=1	4, No

 $[\]rightarrow$ suite des configurations de **c** parcourues : $1 \rightarrow 2 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 14$

<u>Question 2 (1 point):</u> Quelle configuration $\bf c$ trouverait la méthode hill-climbing? Réponse :hill-climbing termine et retourne la configuration $\bf c$ = 14 :

1	2	3
8		4
7	6	5

❖ Si la configuration c=8 est choisie à l'itération n=2 ,au lieu de la configuration c=7

★ <u>n=3:</u>

	ac	tuelle,	n=3			pos	ssibles	s (c'e	est-à-c	dire ni	veau su	iiva	ant),N	=4	
		c=8			(c=9 1	?			c=10	?			c=11	?
Configuration	7	8 1 6	3 4 5		2 1 7	8	3 4 5		2 7	8 1 6	3 4 5		2 7	8 1 6	3 4 5
f(c)		3+5=8			4	1+5=	9		4	1+5= 9	V			4+4=	8

[→] suite des configurations de **c** parcourues : 1→2→8

★ <u>n=4</u>:

	а	ctuelle	e, n=4			р	ossible	es (c'e	st-à	-dire n	iveau	suivar	nt),n=5	1
		c=1	0			ı	c=12 '	?				c=13	?	
Configuration	2 7	8 1 6	3 4 5		_	7	8 1 6	3 4 5			827	1 6	3 4 5	
f(c)		4+5=9				5	5+5=1	0			5-	+5=10	0 V	

[→] suite des configurations de **c** parcourues : $1 \rightarrow 2 \rightarrow 8 \rightarrow 10$

★ <u>n=5</u>:

	а	ctuelle	e, n=5		ро	ossible	s (c'	est-à-c	dire ni	veau sui	/ant),n=	6	
		c=1	3		c=14	?			c=15	?		ı	c=16	?
Configur ation	8 2 7	1 6	3 4 5	2	8 1 6	3 4 5		8 2 7	3 1 6	4 5	2		6	3 4 5
f(c)		5+5=	=10		6+5=	11			6+4=	10		6+	-5=11	V

[→] suite des configurations de **c** parcourues : $1 \rightarrow 2 \rightarrow 8 \rightarrow 10 \rightarrow 13$

★ <u>n=6</u>:

		actu	ielle,	n=6				рс	s	sibles	s (c'e	st-à-di	e	niv	eau sı	uivant),n=6		
			c=16	3		C	=17	?		(=18	?			c=19 '	?		c=20	?
					1			T 1		-					1			1	
		8	1	3		8		3		8	1	3	8	}	1	3	8	1	3
Configur ation		2		4		2	1	4			2	4	2		6	4	2	4	
ation		7	6	5		7	6	5		7	6	5	7	,		5	7	6	5
	,				-											•			· · · · · · · · · · · · · · · · · · ·
f(c)		6-	6+5=11		6+5	5=11	, No		6+5	5=11	, No	()+4	4=10	No	6+	4=10	, No	

[→] suite des configurations de **c** parcourues : $1 \rightarrow 2 \rightarrow 8 \rightarrow 10 \rightarrow 13 \rightarrow 16$

<u>Question 2 (1 point):</u> Quelle configuration **c** trouverait la méthode hill-climbing? Réponse :hill-climbing termine et retourne la configuration c= 16 :

8	1	3
2		4
7	6	5

Exercice 2: MiniMax (3 pts)

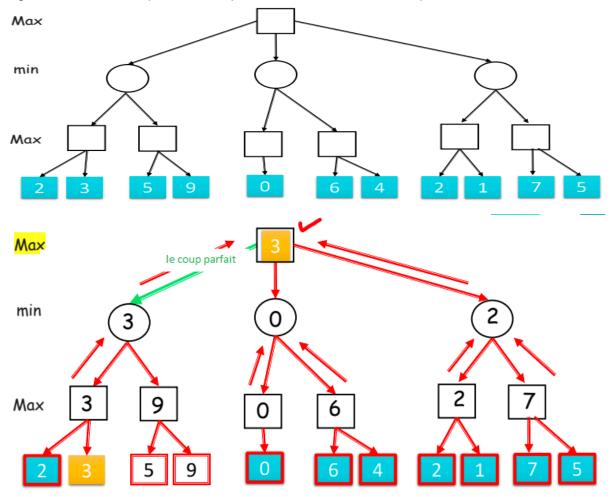
Question 1 (1 point): Compléter les phrases suivantes à l'aide des mots choisis dans la liste ci-dessous.

• la racine, terminal, profondeur d'abord, le plus court chemin, le coup parfait, largeur d'abord, déterministe, admissible, discret

Un jeu à deux joueurs est défini classiquement comme un arbre qui a comme noeuds des positions. Chaque noeud est un noeud «joueur» ou un noeud «opposant».

- > Si un noeud n'a pas de fils, c'est un noeud terminal
- ➤ l'objectif de la méthode MiniMax est de trouver <u>le coup parfait</u> pour un jeu <u>déterministe</u> à information parfaite.
- > l'algorithme MiniMax utilise la recherche en profondeur d'abord

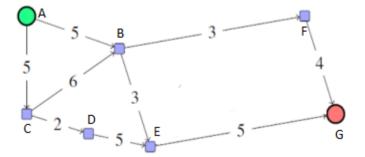
Question 2 (2 points): donner la trace d'exécution (état final seulement) de l'agorithme MiniMax pour l'exemple suivant. Quelle résultat peut-on déduire ?



→ On peut en déduire que le coup parfait pour le joueur Max est vers la gauche.

Exercice 3: A* pour la recherche dans un graphe (11 points)

Considérez la carte suivante. L'objectif est de trouver le chemin le plus court de A vers G. On donne également trois heuristiques, h1, h2 et h3.



Noeud	Α	В	С	D	Е	F	G
h1	10	5	10	10	5	3	0
h2	10	8	11	6	2	5	0
h3	10	6	11	9	2	4	0

Soit la classe Graph suivante :

Questions:

- 1. Donner les instructions qui permettent :
 - de créer les listes open et closed.(0.5 point)

```
→ open = []
closed = []
```

 de créer le nœud de départ et le nœud objectif, sachant que le constructeur def __init__(self, name:str, parent:str), est utilisé pour créer un objet Node. (1 point)

```
→ start_node = Node(start, None)

goal_node = Node(end, None)
```

- d'jouter le nœud de départ à la liste open.(0.5 point)
 - → open.append(start node)
- de créer un objet Graph. (0.5 point)

```
→ graph = Graph()
```

- de créer les connexions de ce graphe (1.5 point,3 corrects ← 0.5 point)

```
→ graph.connect('A', 'B', 5)
graph.connect('A', 'C', 5)
graph.connect('C', 'B', 6)
graph.connect('C', 'D', 2)
graph.connect('B', 'E', 3)
graph.connect('B', 'F', 3)
graph.connect('D', 'E', 5)
graph.connect('F', 'G', 4)
graph.connect('E', 'G', 5)
```

- de créer les heuristiques de h1 pour chaque node. (1 point)

```
→ heuristics = {}
heuristics['A'] = 10
heuristics['B'] = 5
heuristics['C'] = 10
heuristics['D'] = 10
heuristics['E'] = 5
heuristics['F'] = 3
heuristics['G'] = 0
```

2. Est-ce que h1, h2 et h3 sont admissibles ? Justifier. (2 points)

Noeud	А	В	С	D	E	F	G
h1	10	5	10	10	5	3	0
h2	10	8	11	6	2	5	0
h3	10	6	11	9	2	4	0
h*	12	7	12	10	5	4	0

→ h1 est admissible puisque pour chaque noeud h1≤h*.

h2 n'est pas admissible : h2(B)>h*(B), h2(F)>h*(F).

h3 est admissible puisque pour chaque noeud h3≤h*.

- Quelles relations de dominance existent entre ces trois heuristiques? (1 point)
 →Pour (h2, h1), on ne peut pas parler de domination puisque h2 n'est pas admissible.
 - →Pour (h2, h3), on ne peut pas parler de domination puisque h2 n'est pas admissible.
 - →Pour (h1, h3), ni l'un ni l'autre puisque on a h3(B)>h1(B) et h1(D)>h3(D).
- 4. Appliquer la recherche A* en utilisant h1. Donner la suite des noeuds développés. (3 points)

	Contenu de open à chaque itération (état, f, parent)	Contenu de closed à chaque itération
1	(A ,10, void)	vide
2	(B ,10, A),(C,15, A)	(A,10, void)
3	(F ,11, B),(E,13, B),(C,15, A)	(A,10, void),(B,10, A)
4	(G ,12, F),(E,13, B),(C,15, A)	(A,10, void),(B,10, A),(F,11, B)
5	Solution : A→B→F→G	(A,10, void),(B,10, A),(F,11, B),(G,12, B)

% Fin %