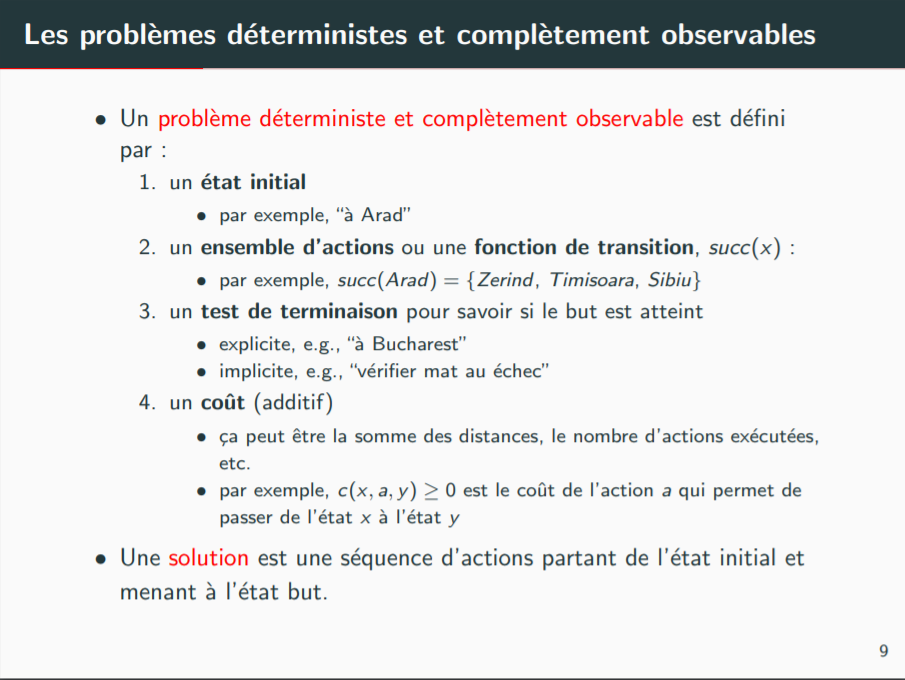
**Intelligence artificielle (IF06M100) • TD 2   
Algorithmes de recherche en IA**



**Exercice 1**

Donnez l’état initial, le but, la fonction successeur et la fonction de coût pour chacun des problèmes suivants.

**Question 1**

Vous devez colorier une carte de façon à ce que les pays adjacents ne soient pas de la même couleur, et en sachant que vous avez à votre disposition 4 couleurs distinctes

|  |  |
| --- | --- |
| Etat initial | Aucune pays coloré |
| Le but | Tous les pays colorés et aucun pays adjacent n'a la même couleur |
| La fonction successeur | Attribuer une couleur à un pays. |
| La fonction de coût | Nombre d'affectations |

**Exercice 1**

Donnez l’état initial, le but, la fonction successeur et la fonction de coût pour chacun des problèmes suivants.

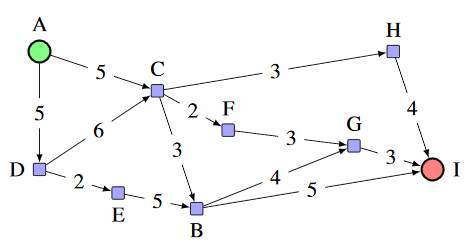
**Question 2**

Un singe mesurant 1 mètre se trouve dans une pièce de 3 mètres de hauteur. Une banane est suspendue au plafond de cette pièce, et le singe aimerait bien avoir cette banane. La pièce contient également 2 caisses qu’il peut déplacer et sur lesquelles il peut monter, chaque caisse mesurant 1 mètre

|  |  |
| --- | --- |
| Etat initial | Comme décrit dans le texte Un singe mesurant 1 mètre se trouve dans une pièce de 3 mètres de hauteur.  Une banane est suspendue au plafond de cette pièce. La pièce contient également 2 caisses, chaque caisse mesurant 1 mètre. |
| Le but | Le singe arrive à avoir la banane |
| La fonction successeur | Montez sur une caisse ; Descendez de la caisse ; Pousser la caisse d'un endroit à un autre, marcher d'un endroit à un autre ; saisir la banane (si debout sur une caisse) |
| La fonction de coût | Nombre d'actions |

**Exercice 2**

Considérez la carte (orientée) suivante.   
L’objectif est de trouver un chemin allant de A à I.



Donner l’ordre de parcours des nœuds pour les algorithmes suivants.

* Si vous avez le choix entre deux nœuds, vous développerez en priorité le premier dans l’ordre alphabétique.   
  On suppose que nous pouvons éviter les répétitions : un état contenu dans un nœud déjà développé ne le sera plus.

1. largeur d’abord

2. coût uniforme

3. profondeur d’abord

**Exercice 3**

Considérez un espace de recherche dans lequel l’état initial est 1   
et la fonction successeur pour un nœud   
retourne deux états contenant les entiers et .

**Question 1**

Dessiner la partie de l’espace de recherche contenant les nœuds de 1 à 15

L’état initial est 1

2n = 2\*1 + 1 = 3

2n = 2\*1 = 2

2n = 2\*3 + 1 = 7

2n = 2\*3 = 6

2n = 2\*2 = 4

2n = 2\*2 + 1 = 5

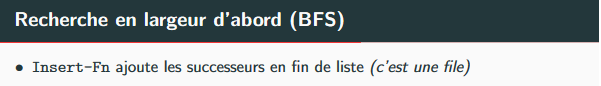
**Exercice 3**

Considérez un espace de recherche dans lequel l’état initial est 1   
et la fonction successeur pour un nœud   
retourne deux états contenant les entiers et .

**Question 2**

Supposez que le but soit 11,   
et que vous considérez l’espace de recherche complet.   
  
Donner l’ordre de parcours des nœuds pour les algorithmes :

(a) largeur d’abord



fringe = [1]  
Une file qui représente l’ensemble des nœud a développer



L’état initial est 1

fringe = [2, 3]

fringe = [3,4,5]

fringe = [4,5,6,7]

fringe = [5,6,7,8,9]

fringe = [7,8,9,10,11,12, 13]

**11 – L’état final**  
on n’a pas encore vérifier que 11 était bien l’état but donc je ne m’arrête pas parce que Je n’ai pas encore développé ces nœuds.

fringe = [8,9,10,11,12, 13,14,15]

fringe = [12, 13,14,15]

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

**Question 2**

Supposez que le but soit 11,   
et que vous considérez l’espace de recherche complet.   
  
Donner l’ordre de parcours des nœuds pour les algorithmes :

(b) profondeur d’abord



fringe = [1]  
Une pile qui représente l’ensemble des nœud à développer



L’état initial est 1

fringe = [2, 3]

fringe = [4,5,3]

fringe = [8,9,5,3]

fringe = [5,3]

fringe = [10,11,3]

**11 – L’état final**  
on n’a pas encore vérifier que 11 était bien l’état but donc je ne m’arrête pas parce que Je n’ai pas encore développé ces nœuds.

fringe = [3]

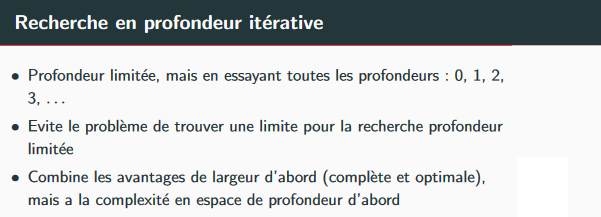
fringe = [12, 13,14,15]

1,2,4,8,9,5,10,11

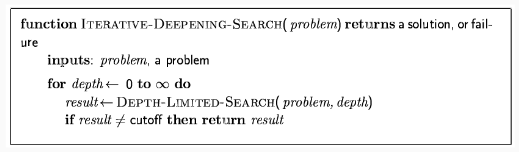
**Question 3**

Supposez que le but soit 11,   
et que vous considérez l’espace de recherche complet.   
  
Donner l’ordre de parcours des nœuds pour les algorithmes :

(c) profondeur itérative



Si le coup des actions est 1.



Recherche en profondeur itérative : l = 0

Nœuds développes : [1]

Nœuds développes : [1 ;1,2,3]  
On a redéveloppé 1

Recherche en profondeur itérative : l = 1

Nœuds développes : [1 ;1,2,3 ; 1,2,4,5,3,6,7]

Recherche en profondeur itérative : l = 2

Recherche en profondeur itérative : l = 3

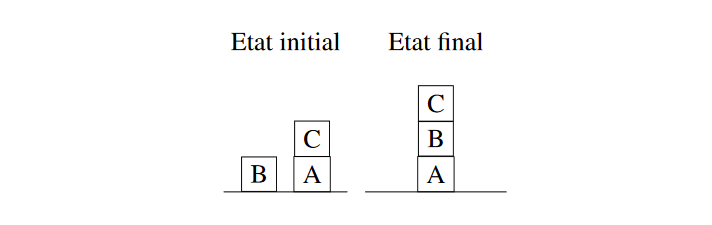
Nœuds développes : [1 ;1,2,3 ; 1,2,4,5,3,6,7 ; 1, 2, 4, 8 , 9 , 5 , 10 ,11]

**Exercice 4**

On est dans la situation initiale donnée sur la figure suivante.

Sur une table sont posés trois cubes, les cubes A et B à même la table et le cube C sur le cube A. On suppose que la position des cubes sur la table les uns par rapport aux autres et indifférencié (que B soit à droite ou à gauche de A ne fait aucune différence).

On veut atteindre l’état final. On a juste le droit de soulever un cube qui n’est pas recouvert par un autre cube et de le reposer ailleurs.



**Question 1**

Dessiner l’espace de recherche de ce problème.

B

C

A

C

A

B

C

B

A

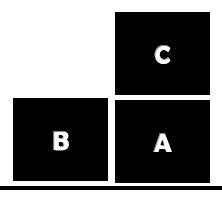
C

B

A

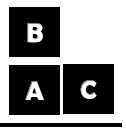
L’état initial



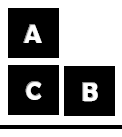




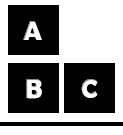




















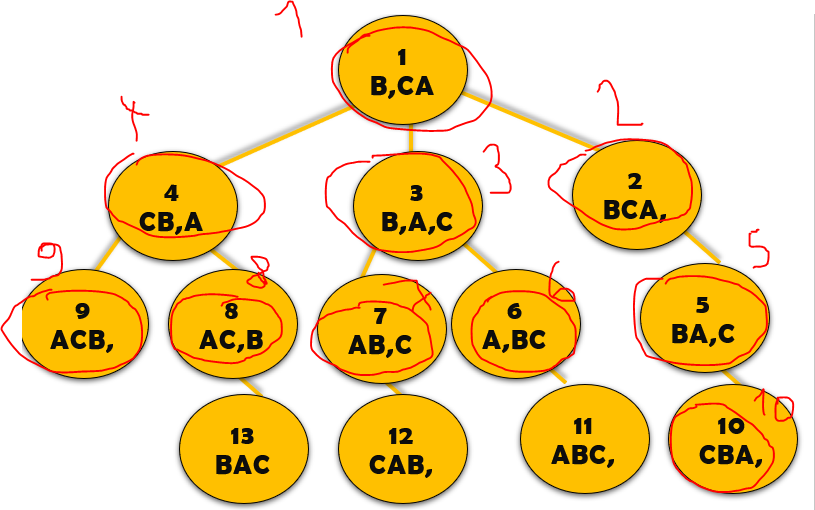
L’état  
 final

**Exercice 4**

**Question 2**

Donner l’ordre de parcours des nœuds pour les algorithmes (on suppose que nous pouvons éviter les répétitions : un état contenu dans un nœud déjà développé ne le sera plus) :

(a) largeur d’abord



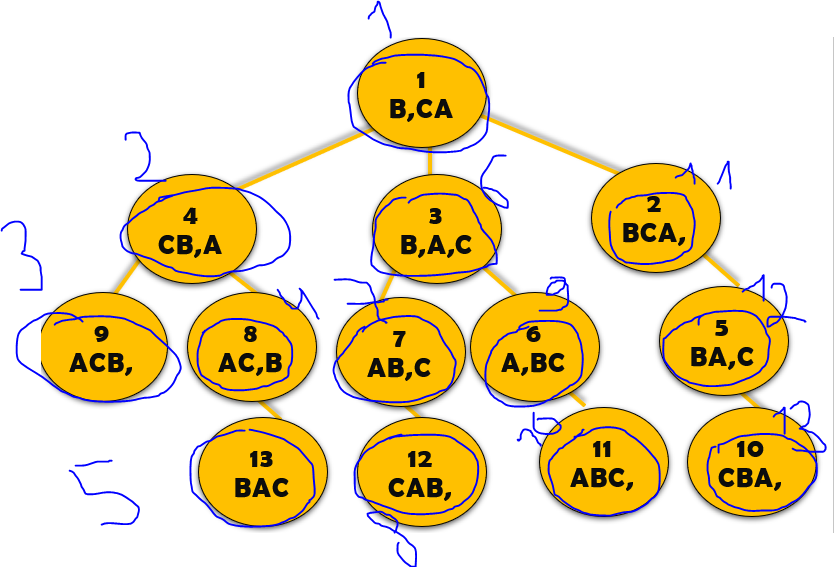
1,2,3,4,5,6,7,8,9,10

**Exercice 4**

**Question 2**

Donner l’ordre de parcours des nœuds pour les algorithmes (on suppose que nous pouvons éviter les répétitions : un état contenu dans un nœud déjà développé ne le sera plus) :

(b) profondeur d’abord



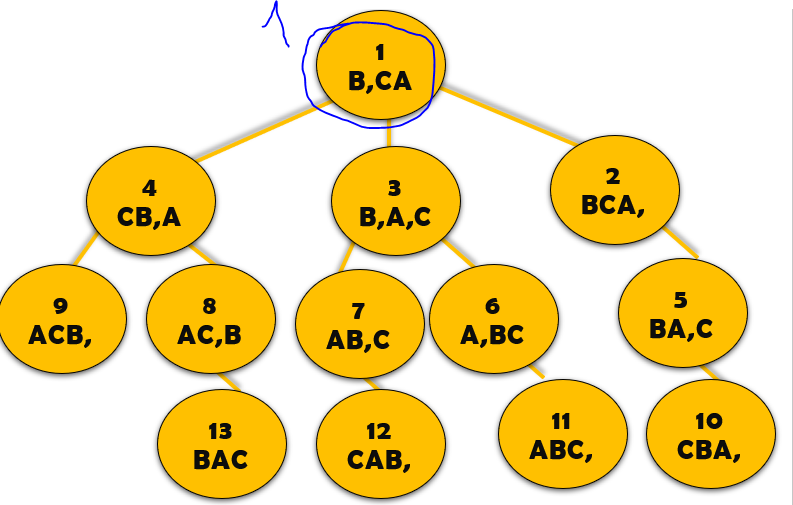
1,4,9,8,13,3,7,12,6,11,2,5,10 [?]

**Question 2**

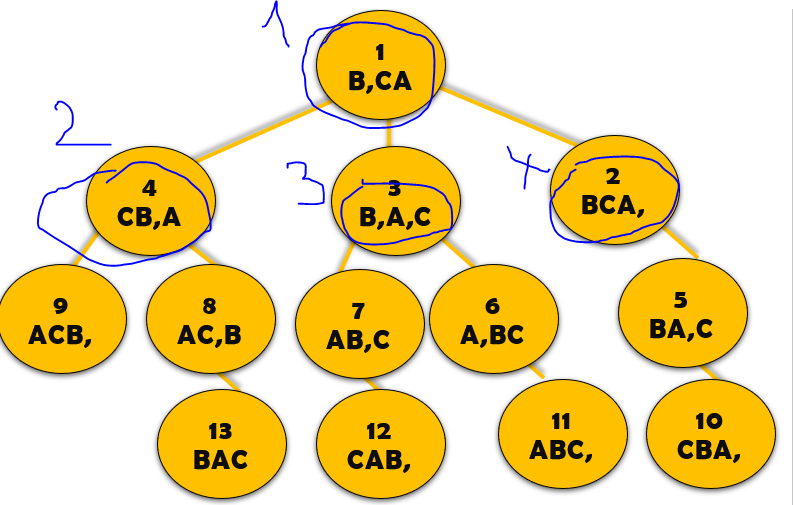
Donner l’ordre de parcours des nœuds pour les algorithmes (on suppose que nous pouvons éviter les répétitions : un état contenu dans un nœud déjà développé ne le sera plus) :

**Exercice 4**

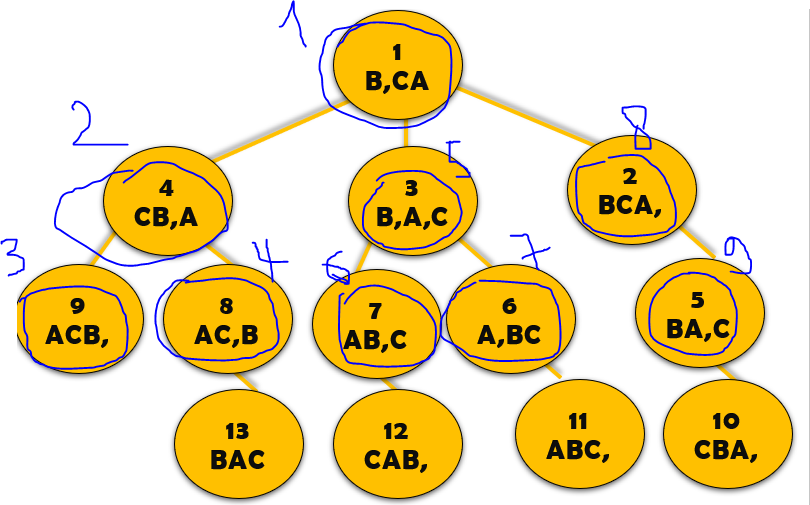
(c) profondeur itérative



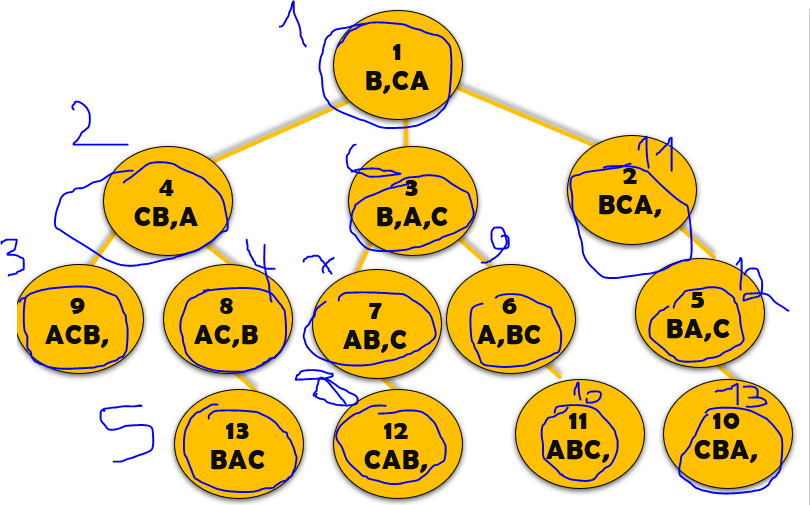
1



1,4,3,2



1,4,9,8,3,7,6,2,5 [?]



1,4,9,8,13,3,7,12,6,11,2,5,10 [?]

***a***

***f***