

UNIVERSITE DE LAUSANNE  
**Faculté des Hautes Etudes Commerciales**

Professeurs	Matière	Session
<b>Benoît Garbinato</b>	<b>Algorithms and Computational Thinking</b>	<b>Hiver</b>

*Durée : 1 heure.*

*À la fin de l'examen, vous rendrez l'ensemble des feuilles que vous avez reçu (énoncé + feuille de réponses). Les deux doivent porter votre nom, prénom et numéro d'étudiant. Pour chaque question, une seule réponse est correcte.*

*Nom et prénom:*

.....  
*No matricule:*

.....

## 1 Questions à choix multiples

### Question 1

Trouvez un arbre couvrant minimal (minimum spanning tree) pour le graphe suivant. Si l'on part de l'arête A, quelle est la hauteur de l'arbre ?

- A [(B, 2), (D, 5), (E, 1)]
- B [(C, 1), (D, 3), (G, 4)]
- C [(G, 4)]
- D [(F, 6)]
- E [(F, 2)]
- F [(G, 2)]

- A 7
- B 4
- C 3
- D 2

### Question 2

Soit la liste d'entiers suivante : [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40].

Laquelle de ces propositions représente une sous-liste qui devra être triée après 3 appels récursifs à l'algorithme de tri par fusion (merge sort) ?

- A [29, 28, 2, 9]
- B [21, 1]
- C [21, 1, 26, 45]
- D [21, 1, 26, 45, 29, 28, 2, 9]

**Question 3**

On veut que le code suivant affiche les nombres de 1 à N qui sont multiples de 7 mais qui ne sont pas multiples de 5. Par quoi faut-il remplacer "#LINE REMOVED" (ligne 4) pour que le code fonctionne ?

```

1 N=100
2 l=[]
3 for i in range(1, N):
4     #LINE REMOVED
5     l.append(str(i))
6
7 print(l)

```

- (A) If  $i/7==0$  and  $i/5==0$ :
- (B) If  $i\%7==0$  and  $i\%5!=0$ :
- (C) If  $i/7==0$  and  $i/5!=0$ :
- (D) If  $i\%7==0$  and  $i\%5==0$ :

**Question 4**

Parmi les affirmations suivantes, laquelle est correct ?

- (A) Une recherche binaire (binary search) a une complexité dans le pire des cas (worst-case complexity) meilleure que la recherche séquentielle (sequential search)
- (B) Une recherche séquentielle (sequential search) a une complexité dans le pire des cas (worst-case complexity) plus faible que la recherche binaire (binary search)
- (C) Une recherche séquentielle (sequential search) a une complexité dans le pire des cas (worst-case complexity) de  $O(N^2)$
- (D) Une recherche séquentielle (sequential search) a une complexité dans le plus faible des cas (best-case complexity) de  $O(N)$

**Question 5**

Soit le code suivant en Python, qu'est-ce qui sera affiché dans la console après son exécution ?

```

1 x = True
2 y = False
3 z = False
4
5 if y or (x and z):
6     print('output1')
7 if not x or (y and z):
8     print('output2')
9 if (x or z) or (y and z):
10    print('output3')
11 if (y != z):
12    print('output4')

```

- (A) output3
- (B) output1
- (C) output4
- (D) output2

**Question 6**

Soit le code suivant en Python, qu'est-ce qui sera affiché dans la console après son exécution ?

```

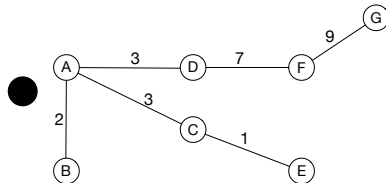
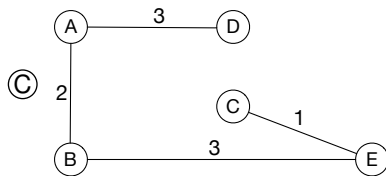
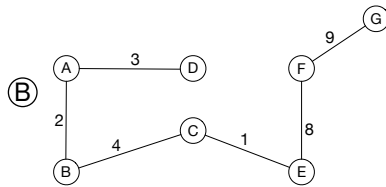
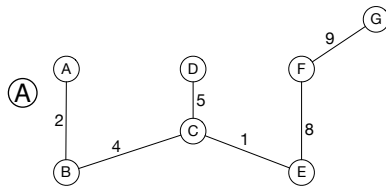
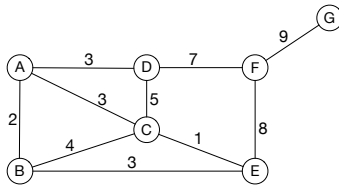
1 age = 20
2 output = ""
3 if age < 18:
4     output += "Vous n'avez pas l'age requis "
5 elif age >= 18:
6     output += "Vous avez l'age requis "
7 elif age <25:
8     output += "Vous avez un rabais "
9 print(output)

```

- (A) Vous n'avez pas l'age requis
- (B) Vous avez l'age requis Vous avez un rabais
- (C) Vous avez un rabais
- (D) Vous avez l'age requis

**Question 7**

Quel est l'arbre couvrant minimal (minimum spanning tree) du graphe ci-dessous ?



**Question 8**

Quelle est la complexité dans le pire des cas (worst-case complexity) du programme python suivant :

```

1 i = N
2 a = 0
3 while i>1:
4     a += i
5     i /= 2
6     print(i)

```

- (A)  $O(N)$   
 (B)  $O(\log(N))$   
 (C)  $O(N^2)$   
 (D)  $O(N/2)$

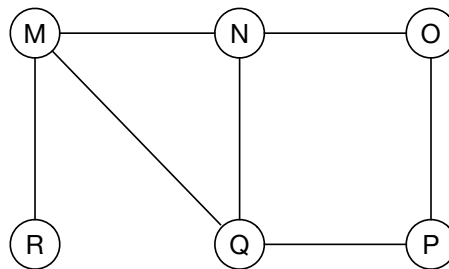
**Question 9**

Identifiez l'ordre de visite des arêtes (vertices) qui est réalisable avec une recherche en largeur (breadth first search) sur le graphe ci-dessous. Si besoin, vous pouvez vous référer à la méthode bfs en Python ci-dessous, qui effectue une recherche en largeur. Notez qu'il est possible de répondre à la question sans regarder le code.

```

1 def bfs(graph, start):
2     explored = list()
3     queue = [start]
4
5     while len(queue) > 0:
6         node = queue.pop(0)
7         if node not in explored:
8             explored.append(node)
9             neighbours = graph[node]
10
11            for neighbour in neighbours:
12                queue.append(neighbour)
13    return explored

```



- (A) NQMPOR  
 (B) MNOPQR  
 (C) QMNPOR  
 (D) QMNPRO

**Question 10**

Soit le code suivant en Python, qu'est-ce qui sera affiché dans la console après son exécution ?

```
print([1, 2, 3]+[3, 2, 1, 0])
```

- (A) [4, 4, 4]
- (B) [4, 4, 4, 0]
- (C) [1, 2, 3, 3, 2, 1, 0]
- (D) [13, 22, 31, 0]

**Question 11**

Supposons que les nombres 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 soient insérés dans cet ordre dans un arbre binaire de recherche initialement vide. Quelle est la séquence de parcours (traversal sequence) de l'arbre ?

```
1 # A function to do inorder tree traversal
2 def printInorder(root):
3     if root:
4         # First recur on left child
5         printInorder(root.left)
6         # then print the data of node
7         print (root.val),
8         # now recur on right child
9         printInorder(root.right)
```

- (A) 0 2 4 3 1 6 5 9 8 7
- (B) 7 5 1 0 3 2 4 6 8 9
- (C) 0 1 2 3 4 5 6 7 8 9
- (D) 9 8 6 4 2 3 0 1 5 7

**Question 12**

Soit le code suivant en Python, qu'est-ce qui sera affiché dans la console après son exécution ?

```
1 def mul(a,b):
2     if b == 0:
3         return 0
4     else:
5         return a + mul(a, b-1)
6 print (mul (2,3))
```

- (A) 4
- (B) 6
- (C) 5
- (D) 2

**Question 13**

Soit le code suivant en Python, qu'est-ce qui sera affiché dans la console après son exécution ?

```
1 vehicules = ["voiture", "velo"]
2 vehicules.append("avion")
3 vehicules[2] = "train"
4 print(vehicules)
```

- (A) ["voiture", "train", "avion"]
- (B) ["voiture", "velo", "avion"]
- (C) ["voiture", "velo", "train"]
- (D) ["voiture", "train"]

**Question 14**

Comment est stocké l'entier (integer) -57 dans un espace d'adressage de 8 bits avec une représentation binaire utilisant le complément à deux (2's complement) ?

- (A) 11000111
- (B) 11000010
- (C) 11000110
- (D) 11000100

**Question 15**

Soit le code suivant en Python, qu'est-ce qui sera affiché dans la console après son exécution ?

```
1 num_int = 456
2 num_string = "456"
3 total_num = num_int + int(num_string)
4 print(type(total_num))
```

- (A) <class 'string'>
- (B) <class 'int'>
- (C) TypeError: must be str, not int
- (D) <class 'float'>

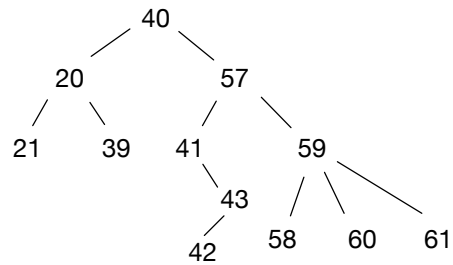
**Question 16**

Dans un arbre binaire équilibré (balanced binary search tree) chaque noeud interne de l'arbre a deux enfants (all internal nodes have 2 children). Les entiers suivants sont insérés dans un arbre binaire équilibré initialement vide dans l'ordre suivant: 10, 1, 3, 5, 15, 12, 16. Quelle est la hauteur de l'arbre binaire équilibré une fois les entiers insérés ?

- (A) 3
- (B) 4
- (C) 2
- (D) 6

**Question 17**

Soit l'arbre ci-dessous, quelle affirmation est correcte ?



- (A) C'est un arbre de recherche binaire (binary search tree)
- (B) C'est un arbre équilibré (balanced tree)
- (C) C'est un arbre binaire (binary tree)
- (D) Aucune des affirmations n'est vraie

**Question 18**

Quelle est la complexité dans le pire des cas (worst-case complexity) du programme Python suivant :

```

1 for i in range(10):
2     i += 1
3     for j in range(N):
4         j += 1
5         for k in range(N-2, N+2):
6             print('hello world')
  
```

- (A)  $O(N)$
- (B)  $O(2^N)$
- (C)  $O(N^3)$
- (D)  $O(N \log(N))$

**Question 19**

Quelle est la complexité dans le pire des cas (worst-case complexity) de la méthode ajout\_element pour une liste de N éléments?

```
1 def tri(liste):
2     triee = False
3     while not triee:
4         triee = True
5         for i in range(len(liste)-1):
6             if liste[i]>liste[i+1]:
7                 triee = False
8                 liste[i], liste[i+1] = liste[i+1], liste[i]
9
10 def ajout_element(liste, element):
11     liste.append(element)
12     tri(liste)
13     return liste
14
15 #Example, the following line will print: [1, 2, 5, 6, 8, 9]
16 print(ajout_element([1,5,2,6,9], 8))
```

- (A)  $O(N \log(N))$
- (B)  $O(N)$
- (C)  $O(N^2)$
- (D)  $O(2^N)$

**Question 20**

Quelle est la chaîne d'étapes suivie par le processeur pour traiter une instruction ?

- (A) fetch → decode → execute → store
- (B) decode → store → execute → fetch
- (C) store → decode → fetch → execute
- (D) fetch → execute → decode → store





### Feuille de réponses

Exemple: n° 12031123

0  0  0  0  0  0  0  
 1  1  1  1  1  1  1  
 2  2  2  2  2  2  2  
 3  3  3  3  3  3  3

← encodez votre numéro d'étudiant ci-contre, et inscrivez votre nom, prénom et numéro de place ci-dessous.

0  0  0  0  0  0  0  0  
 1  1  1  1  1  1  1  1  
 2  2  2  2  2  2  2  2  
 3  3  3  3  3  3  3  3  
 4  4  4  4  4  4  4  4  
 5  5  5  5  5  5  5  5  
 6  6  6  6  6  6  6  6  
 7  7  7  7  7  7  7  7  
 8  8  8  8  8  8  8  8  
 9  9  9  9  9  9  9  9

Nom: .....

Prénom: .....

Place: .....

**Les réponses aux questions sont à donner exclusivement sur cette feuille : les réponses données sur d'autres feuilles ne seront pas prises en compte.**

**Coloriez complètement et proprement chaque case choisie, afin d'obtenir le meilleur contraste possible par rapport à une case vide. Ne coloriez pas les cases au crayon à papier, n'utilisez pas de couleur rouge (le programme correcteur ne voit pas les couleurs). Utilisez plutôt du noir ou du bleu foncé. Pour corriger une éventuelle erreur de saisie, veuillez utiliser si possible un correcteur liquide (Tipp-Ex).**

### 1 Questions à choix multiples

QUESTION 1 :  A  B  C  D  
 QUESTION 2 :  A  B  C  D  
 QUESTION 3 :  A  B  C  D  
 QUESTION 4 :  A  B  C  D  
 QUESTION 5 :  A  B  C  D  
 QUESTION 6 :  A  B  C  D  
 QUESTION 7 :  A  B  C  D  
 QUESTION 8 :  A  B  C  D  
 QUESTION 9 :  A  B  C  D

QUESTION 10 :  A  B  C  D  
 QUESTION 11 :  A  B  C  D  
 QUESTION 12 :  A  B  C  D  
 QUESTION 13 :  A  B  C  D  
 QUESTION 14 :  A  B  C  D  
 QUESTION 15 :  A  B  C  D  
 QUESTION 16 :  A  B  C  D  
 QUESTION 17 :  A  B  C  D  
 QUESTION 18 :  A  B  C  D  
 QUESTION 19 :  A  B  C  D  
 QUESTION 20 :  A  B  C  D