Nom :

Numéro de votre place :

Code permanent :

Directives pédagogiques :

- Inscrivez votre nom, numéro de place et code permanent.
- Sortez votre carte étudiante et mettez la à vue.
- Lisez attentivement toutes les questions et **répondez directement sur le questionnaire**.
- Seule l'utilisation d'un crayon est permise, **aucune documentation, calculatrice, téléphone cellulaire, ordinateur, ou autre objet**.
- Cet examen contient 25 questions de 4 points chacune, pour un total de 100 points.
- Cet examen contient 17 pages, dont 4 pages brouillon que vous pouvez détacher.
- Pour les questions à développement, **écrivez lisiblement et détaillez vos réponses**.
- Vous avez 165 minutes pour compléter cet examen.

BONNE CHANCE ET BON ÉTÉ!

1	/ 4	11	/ 4	21	/ 4
2	/ 4	12	/ 4	22	/ 4
3	/ 4	13	/ 4	23	/ 4
4	/ 4	14	/ 4	24	/ 4
5	/ 4	15	/ 4	25	/ 4
6	/ 4	16	/ 4		
7	/ 4	17	/ 4		
8	/ 4	18	/ 4		
9	/ 4	19	/ 4		
10	/ 4	20	/ 4		
Total	/ 40	Total	/ 40	Total	/ 20
				Total	/100

Question 1 Que sera-t-il affiché lors de l'exécution du code Python suivant? :

```
x = [2*i for i in range(-32, 32, 2)]
print(x[-1])
```

Question 2 Que sera-t-il affiché lors de l'exécution du code Python suivant? :

```
def scale1(data, factor):
    for j in range(len(data)):
        data[j] *= factor
data = [i for i in range (4)]
scale1(data, 3)
print(data)
```

Question 3 Que sera-t-il affiché lors de l'exécution du code Python suivant? :

```
def scale2(data, factor):
    for val in data:
        val *= factor
data = [i for i in range (4)]
scale2(data, 3)
print(data)
```

Question 4 Que sera-t-il affiché lors de l'exécution du code Python suivant? :

```
def B(data):
    x = float('-inf')
    y = float('-inf')
    for i in data:
        if i > y:
            x, y = y, i
        elif i > x:
            x = i
    return (x, y)
data = [4, -8, 0, 3000, -1234, 45, 3, -6, -100]
print(str(B(data)))
```

Quelques rappels en Python :
1) lors d'une somme de booléens, ceux-ci sont convertis en entiers (False $= 0$ et True $= 1$)
2) bool(i) $=$ False si $i = 0$ et True sinon
3) l'évaluation d'une clause comme (A and B) ou (A or B) est paresseuse : si le 1er booléen est suffisant, le 2e n'est pas évalué

Lorsque l'on vous demande un nombre de comparaisons, il doit être en fonction de n.

Question 5 Soit la fonction récursive suivante :

```
def present_1(data, i, element):
    if i == len(data):
        return False
    else:
        return ( (data[i] == element) or present_2(data, i+1, element) )
```

Lorsqu'initialement appelée avec un 1er argument de longueur $n > 0$ et un 2e argument égal à 0:

Que fait cette fonction?

Combien de comparaisons ($==$) effectue-t-elle dans le **pire** cas?

Combien de comparaisons ($==$) effectue-t-elle dans le **meilleur** cas?

Question 6 Soit la fonction récursive suivante :

```
def present_2(data, i, element):
    if i == len(data):
        return False
    else:
        return bool( (data[i] == element) + present_1(data, i+1, element) )
```

Lorsqu'initialement appelée avec un 1er argument de longueur $n > 0$ et un 2e argument égal à 0:

La fonction `present_2` retourne-t-elle le même résultat que la fonction `present_1`?

Combien de comparaisons ($==$) effectue-t-elle dans le **pire** cas?

Combien de comparaisons ($==$) effectue-t-elle dans le **meilleur** cas?

Important
(1) Sauf si spécifié, toutes les questions portent sur des structures de données contenant $\boldsymbol{n}$ **éléments**.
(2) Lorsqu'on vous demande une complexité, on sous-entend **dans le pire cas**.
(3) Lorsqu'on vous demande une complexité, vous devez donner l'ensemble. Par exemple, si la réponse est $\mathcal{O}(n)$, **vous n'aurez pas de point pour** $\boldsymbol{n}$.
(4) Lorsqu'on vous demande une complexité $\mathcal{O}$, vous devez donner la **plus petite borne** supérieure. De même, lorsque l'on vous demande une complexité Ω, vous devez donner la **plus grande borne** inférieure. **Aucun point ne sera accordé pour un ensemble plus grand.** Par exemple, si la réponse est $\mathcal{O}(n)$, vous n'aurez pas de point pour $\mathcal{O}(n!)$.
(5) Lorsqu'on vous demande une complexité, vous devez donner la plus simple expression qui la décrit. Par exemple, si la réponse est $\mathcal{O}(n^2)$, **vous n'aurez pas de point pour** $\boldsymbol{\mathcal{O}(6n^2 + 7n + 42)}$.

Question 7 Soit une **Liste**, implantée en tableau **dynamique** (*dynamic array*).

Quelle est sa complexité $\mathcal{O}$ en espace?

Quelle est sa complexité Ω en espace?

Quelle est la complexité $\mathcal{O}$ en temps de l'accès à un élément par index?

Quelle est la complexité $\mathcal{O}$ en temps de l'insertion d'un élément à un index quelconque?

Quelle est la complexité $\mathcal{O}$ en temps (amorti) de l'ajout d'un élément à la fin de la liste (*append*)?

Question 8 Soit une **Pile**, implantée en tableau **dynamique** (*dynamic array*).

Quelle est sa complexité $\mathcal{O}$ en espace?

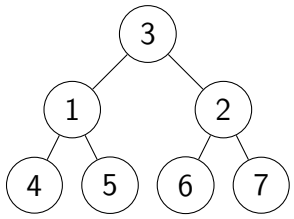
Quelle est la complexité $\mathcal{O}$ en temps (amorti) de l'empilement d'un élément sur une pile (*push*)?

Quelle est la complexité $\mathcal{O}$ en temps (amorti) de l'empilement de n éléments sur une pile initialement vide?

Question 9 Soit une **Liste doublement chaînée**.

Quelle est la complexité $\mathcal{O}$ en temps pour accéder à l'élément de la queue (*tail*)?

Quelle est la complexité $\mathcal{O}$ en temps de la délétion d'un élément sur lequel un pointeur est donné?

Question 10 Dans quel ordre seraient visités les noeuds de l'arbre 

lors d'un parcours :

pré-ordre (*pre-order*)?

en-ordre (*in-order*)?

post-ordre (*post-order*)?

en largeur (*breadth-first*)?

Question 11 Soit un arbre binaire dont les parcours suivants visiteraient les clés cet ordre :
en-ordre (*in-order*) : H T U R A D R N E T
post-ordre (*post-order*) : H U T R D N T E R A
Dans quel ordre seraient visitées les clés lors d'un parcours :

en largeur (*breadth-first*)?

pré-ordre (*pre-order*)?

Question 12 On veut trier les nombres contenus dans le tableau suivant à l'aide du tri par monceau *in-place*. Remplissez le tableau suivant, qui doit représenter l'état du tableau après la phase I du tri, soit après la construction du monceau. Vous n'obtiendrez les points que si le tableau est exact.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	0

Question 13 Soit un monceau encodé dans un tableau débutant à l'indice 0.

a) À quel(s) indice(s) dans le tableau du monceau peut se trouver la 10$^{\text{e}}$ plus petite clé?

b) Si la clé k peut se trouver à n'importe quel indice i tel que $1 \leq i \leq 62$, quel serait le rang de k si les clés étaient triées ($1^{\text{er}}, 2^{\text{e}}, 3^{\text{e}}, ...$)?

Question 14 Remplissez le tableau de hachage de longueur 11, initialement vide, après l'insertion, dans cet ordre, des clés 5, 11, 12, 13, 16, 20, 23, 39, 44, 88 et 94 en utilisant la fonction de hachage $h(i) = (3i - 5) \bmod 11$ et en assumant que les collisions sont traitées par sondage linéaire.

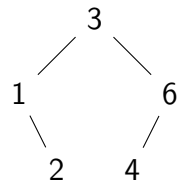
0	1	2	3	4	5	6	7	8	9	10

Question 15 Remplissez le tableau de hachage de longueur 11, initialement vide, après l'insertion, dans cet ordre, des clés 94, 88, 44, 39, 23, 20, 16, 13, 12, 11 et 5 en utilisant la fonction de hachage $h(i) = (3i - 5) \bmod 11$ et en assumant que les collisions sont traitées par double hachage en utilisant la fonction de hachage secondaire $h'(k) = 7 - (k \bmod 7)$.

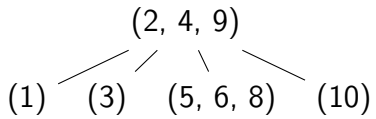
0	1	2	3	4	5	6	7	8	9	10

Question 16

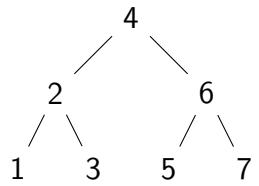
Une fonction dans l'ensemble $\mathcal{O}(n^2)$ est nécessairement dans l'ensemble $\mathcal{O}(n)$.	VRAI	FAUX
La(s)quelle(s) de ces deux structures de données est habituellement utilisée pour accumuler des appels récursifs?	une file	une pile
Quelle est la plus petite borne supérieure de l'ordre de complexité pour interchanger deux noeuds d'une liste doublement chaînée contenant n noeuds, sur lesquels deux pointeurs sont donnés, dans le pire cas?	$\mathcal{O}(1)$	$\mathcal{O}(n)$
Le(s)quel(s) de ces deux types de parcours garanti(ssen)t de parcourir les noeuds d'un arbre binaire de recherche dans un ordre trié?	*in-order*	en largeur
Une table de hachage permet le parcours de ses n clés en ordre trié en $\mathcal{O}(n)$.	VRAI	FAUX
La recherche d'une clé dans un arbre AVL, un *Spay tree* ou un arbre (2, 4) contenant n clés est garantie en $\mathcal{O}(n \lg n)$.	VRAI	FAUX
Quelle est la plus petite borne supérieure de l'ordre de complexité du tri par monceau (*heap-sort*) de n éléments, dans le pire cas?	$\mathcal{O}(n \lg n)$	$\mathcal{O}(n^2)$
Une fonction dans l'ensemble $\mathcal{O}(n)$ est nécessairement dans l'ensemble $\mathcal{O}(n^2)$.	VRAI	FAUX



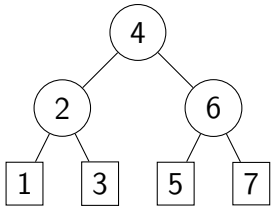
Question 17 Re-dessinez l'arbre AVL
après l'**insertion** de la **clé 5** :



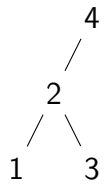
Question 18 Re-dessinez l'arbre (2, 4)
après l'**insertion** de la **clé 7** :



Question 19 Re-dessinez le *Splay Tree*
après la **suppression** de la **clé 7** :

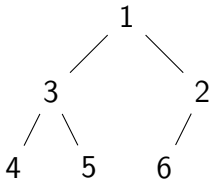


Question 20 Re-dessinez l'arbre rouge-noir
après l'**insertion** de la **clé 8** ($\bigcirc$ = noir, $\square$ = rouge) :

Question 21 Encerclez la bonne réponse. L'arbre 

peut-il être :

Un arbre binaire?	OUI	NON
Un monceau?	OUI	NON
Un arbre binaire de recherche?	OUI	NON
Un arbre AVL?	OUI	NON
Un arbre (2, 4)?	OUI	NON
Un arbre *Splay Tree*?	OUI	NON

Question 22 Encerclez la bonne réponse. L'arbre 

peut-il être :

Un arbre binaire?	OUI	NON
Un monceau?	OUI	NON
Un arbre binaire de recherche?	OUI	NON
Un arbre AVL?	OUI	NON
Un arbre (2, 4)?	OUI	NON
Un arbre *Splay Tree*?	OUI	NON

Question 23 Soit le table de programmation dynamique suivante pour trouver une plus longue sous-séquence commune à deux chaînes de caractères :

			m	y	m	m	e	c	a	c	e	c	o	i	t	
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
t	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
l	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
y	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1
m	3	0	0	1	1	1	1	1	1	1	1	1	1	1	1
m	4	0	1	1	2	2	2	2	2	2	2	2	2	2	2
e	5	0	1	1	2	3	3	3	3	3	3	3	3	3	3
p	6	0	1	1	2	3	4	4	4	4	4	4	4	4	4
f	7	0	1	1	2	3	4	4	4	4	4	4	4	4	4
c	8	0	1	1	2	3	4	4	4	4	4	4	4	4	4
u	9	0	1	1	2	3	4	5	5	5	5	5	5	5	5
p	10	0	1	1	2	3	4	5	5	5	5	5	5	5	5
i	11	0	1	1	2	3	4	5	5	5	5	5	5	5	5
t	12	0	1	1	2	3	4	5	5	5	5	5	5	6	6
	13	0	1	1	2	3	4	5	5	5	5	5	5	6	7

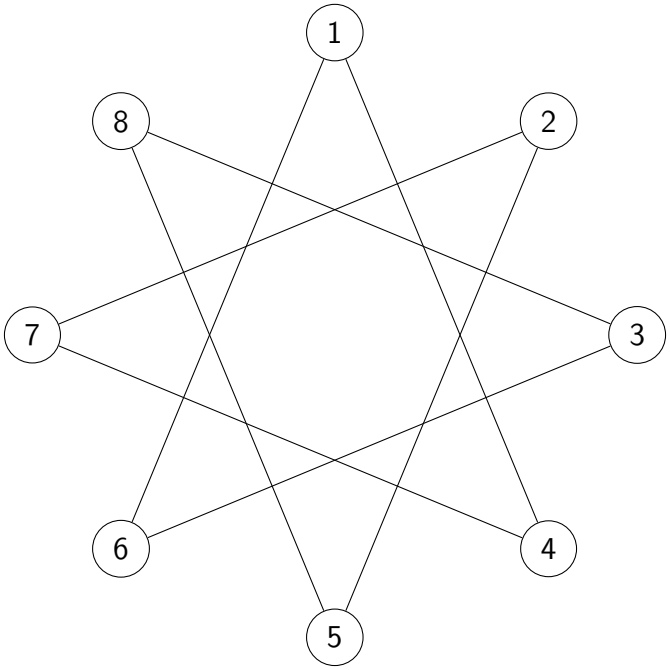
a) Quelle est la plus longue sous-séquence commune?

b) Noircissez dans le tableau le chemin parcouru pour obtenir cette plus longue sous-séquence commune.

Question 24 Dessinez le trie standard contenant les chaînes de caractères suivantes (conservez l'ordre alphabétique des enfants) :

```
{ arbre, trie, arc, tree, arete, cycle, clique }
```

Question 25 Soit le graphe suivant :



En respectant l'ordre croissant des noeuds adjacents, dans quel ordre seront visités les noeuds du graphe suivant, en partant du noeud 7 :

a) lors d'un parcours en profondeur? (une seule réponse possible)

b) lors d'un parcours en largeur? (une seule réponse possible)

Brouillon :

Brouillon :

Brouillon :

Brouillon :