

## INF3105 – 2019A / Examen 1 (27 octobre 2019) / Partie A

<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	2	<input type="checkbox"/>	3	<input type="checkbox"/>	4	<input type="checkbox"/>	5	<input type="checkbox"/>	6	<input type="checkbox"/>	7	<input type="checkbox"/>	8	<input type="checkbox"/>	9
<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	2	<input type="checkbox"/>	3	<input type="checkbox"/>	4	<input type="checkbox"/>	5	<input type="checkbox"/>	6	<input type="checkbox"/>	7	<input type="checkbox"/>	8	<input type="checkbox"/>	9
<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	2	<input type="checkbox"/>	3	<input type="checkbox"/>	4	<input type="checkbox"/>	5	<input type="checkbox"/>	6	<input type="checkbox"/>	7	<input type="checkbox"/>	8	<input type="checkbox"/>	9
<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	2	<input type="checkbox"/>	3	<input type="checkbox"/>	4	<input type="checkbox"/>	5	<input type="checkbox"/>	6	<input type="checkbox"/>	7	<input type="checkbox"/>	8	<input type="checkbox"/>	9
<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	2	<input type="checkbox"/>	3	<input type="checkbox"/>	4	<input type="checkbox"/>	5	<input type="checkbox"/>	6	<input type="checkbox"/>	7	<input type="checkbox"/>	8	<input type="checkbox"/>	9
<input type="checkbox"/>	0	<input type="checkbox"/>	1	<input type="checkbox"/>	2	<input type="checkbox"/>	3	<input type="checkbox"/>	4	<input type="checkbox"/>	5	<input type="checkbox"/>	6	<input type="checkbox"/>	7	<input type="checkbox"/>	8	<input type="checkbox"/>	9

← Entrez les 6 premiers chiffres de votre code permanent. Exemple: ABCD01029211 ==> 010292.

Nom et prénom :

.....  
 .....

Remplissez les cases correspondant aux bonnes réponses. Les questions marquées d'un ♣ peuvent avoir zéro, une ou plusieurs bonnes réponses. Chaque question vaut 2 points pour un total de 60 points.

**Question 1 ♣** Cochez les mots qui sont des **mots-clés** (*keywords* ou mots réservés) du langage C/C++.

<input checked="" type="checkbox"/>	return	<input type="checkbox"/>	sqrt	<input type="checkbox"/>	string
<input checked="" type="checkbox"/>	delete	<input checked="" type="checkbox"/>	while	<input checked="" type="checkbox"/>	unsigned
<input checked="" type="checkbox"/>	new	<input type="checkbox"/>	cin	<input type="checkbox"/>	istream
<input checked="" type="checkbox"/>	short	<input checked="" type="checkbox"/>	char	<input checked="" type="checkbox"/>	do

**Question 2 ♣** En C/C++, la signature d'une fonction est définie par son :

<input checked="" type="checkbox"/>	nombre de paramètres.
<input type="checkbox"/>	type de retour.
<input checked="" type="checkbox"/>	type des paramètres.
<input checked="" type="checkbox"/>	nom.

**Question 3 ♣** Cochez le ou les énoncés illégaux (ayant au moins une erreur).

<input type="checkbox"/>	int *p, *q, **r=&q;
<input type="checkbox"/>	string n, *p=0;
<input type="checkbox"/>	int p; int& q=p;
<input checked="" type="checkbox"/>	double k; int* p = &k;

**Question 4** Le programme progA.cpp laisse \_\_\_\_\_ objets de type int non libérés sur le tas (*heap*).

<input type="checkbox"/>	7	<input type="checkbox"/>	16	<input type="checkbox"/>	2	<input type="checkbox"/>	64
<input type="checkbox"/>	8	<input type="checkbox"/>	0	<input type="checkbox"/>	128	<input checked="" type="checkbox"/>	24
<input type="checkbox"/>	6	<input type="checkbox"/>	5	<input type="checkbox"/>	36	<input type="checkbox"/>	4

**Question 5** Le programme progA.cpp affiche sur la première ligne (fonction f1):

<input type="checkbox"/>	P12P34G_r_p12p34p12p34
<input type="checkbox"/>	P12P34G_r_p34p12p43p21
<input checked="" type="checkbox"/>	P12P34G_r_p34p12r_p34p12p34p12
<input type="checkbox"/>	P21P34G_r_p43p21r_p12p34p12p34
<input type="checkbox"/>	P12P34G_P00P00R_r_p43p21r_p43p21p43p21

**Question 6** Le programme progA.cpp affiche sur la deuxième ligne (fonction f2):

<input type="checkbox"/>	R_P00P00R_P00P00G_P34P12p34p12p12p34r_
<input type="checkbox"/>	P00P00R_P00P00R_P34P12G_r_p12p12p34p34
<input checked="" type="checkbox"/>	P00P00R_P00P00R_P34P12G_r_p34p12p12p34
<input type="checkbox"/>	R_P00P00R_P00P00G_P34P12p12p12p34p34r_
<input type="checkbox"/>	P00P00R_P00P00R_

**Question 7** L'exécution de ./progB < test1.txt affiche sur la première ligne :

<input type="checkbox"/>	d	<input type="checkbox"/>	e	<input type="checkbox"/>	h	<input type="checkbox"/>	a
<input type="checkbox"/>	x	<input type="checkbox"/>	f	<input type="checkbox"/>	k	<input checked="" type="checkbox"/>	c
<input type="checkbox"/>	b	<input type="checkbox"/>	g	<input type="checkbox"/>	y	<input type="checkbox"/>	j

**Question 8** L'exécution de ./progB < test2.txt affiche sur la deuxième ligne :

<input type="checkbox"/>	1	<input type="checkbox"/>	6	<input type="checkbox"/>	5	<input type="checkbox"/>	0
<input type="checkbox"/>	4	<input type="checkbox"/>	8	<input type="checkbox"/>	15	<input type="checkbox"/>	7
<input type="checkbox"/>	18	<input type="checkbox"/>	10	<input type="checkbox"/>	2	<input checked="" type="checkbox"/>	3

**Question 9** Quelle est la complexité temporelle (pire cas) du programme progB.cpp ? Supposez  $n$  phrases,  $m$  mots par phrase et  $k$  mots différents.

<input type="checkbox"/>	$\mathcal{O}(nmk^2)$	<input type="checkbox"/>	$\mathcal{O}(n \log m \log k)$
<input type="checkbox"/>	$\mathcal{O}(n \log k)$	<input type="checkbox"/>	$\mathcal{O}(nk \log m)$
<input type="checkbox"/>	$\mathcal{O}(nmk)$	<input type="checkbox"/>	$\mathcal{O}(nmk \log k)$
<input checked="" type="checkbox"/>	$\mathcal{O}(nm + n \log k)$	<input type="checkbox"/>	$\mathcal{O}(nm \log k)$

**Question 10 ♣** Cochez les expressions de complexité grand-O qui sont simplifiées. Notez que  $k$ ,  $m$  et  $n$  sont des variables indépendantes.

<input checked="" type="checkbox"/>	$\mathcal{O}(n + k)$	<input checked="" type="checkbox"/>	$\mathcal{O}(n^2 + m)$
<input type="checkbox"/>	$\mathcal{O}(n^2 + 8m)$	<input type="checkbox"/>	$\mathcal{O}(3n + 2m)$
<input checked="" type="checkbox"/>	$\mathcal{O}(n!2^n)$	<input type="checkbox"/>	$\mathcal{O}(n + 7)$

**Question 11 ♣** Cochez les énoncés vrais. Les symboles < et > signifient moins et plus complexe que.

<input checked="" type="checkbox"/>	$\mathcal{O}(n!) > \mathcal{O}(2^n)$	<input type="checkbox"/>	$\mathcal{O}(3n) > \mathcal{O}(2n)$
<input checked="" type="checkbox"/>	$\mathcal{O}(2^n) > \mathcal{O}(n^5)$	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	$\mathcal{O}(5n^2 + 9n) < \mathcal{O}(n^3)$	<input type="checkbox"/>	

**Question 12** L'ajout à la fin d'un tableau dynamique a une complexité temporelle amortie de \_\_\_\_\_ lorsque la politique d'agressissement augmente la capacité de 1.

<input type="checkbox"/>	$\mathcal{O}(\log n)$	<input type="checkbox"/>	$\mathcal{O}(1)$
<input type="checkbox"/>	$\mathcal{O}(n \log n)$	<input type="checkbox"/>	$\mathcal{O}(n^2)$
<input checked="" type="checkbox"/>	$\mathcal{O}(n)$	<input type="checkbox"/>	

**Question 13** L'ajout à la fin d'un tableau dynamique a une complexité temporelle amortie de \_\_\_\_\_ lorsque la politique d'agressissement double la capacité.

<input type="checkbox"/>	$\mathcal{O}(\log n)$	<input type="checkbox"/>	$\mathcal{O}(n \log n)$
<input checked="" type="checkbox"/>	$\mathcal{O}(1)$	<input type="checkbox"/>	$\mathcal{O}(n)$
<input type="checkbox"/>		<input type="checkbox"/>	$\mathcal{O}(n^2)$

**Question 14** L'insertion dans un arbre binaire de recherche équilibré a une complexité temporelle de :

<input type="checkbox"/>	$\mathcal{O}(n)$	<input type="checkbox"/>	$\mathcal{O}(n \log n)$
<input checked="" type="checkbox"/>	$\mathcal{O}(\log n)$	<input type="checkbox"/>	$\mathcal{O}(n^2)$
<input type="checkbox"/>		<input type="checkbox"/>	$\mathcal{O}(1)$

**Question 15** Le test d'équivalence (operator ==) pour un arbre binaire de recherche **équilibré** a une complexité temporelle de \_\_\_\_\_. Supposez la meilleure implémentation possible de cet opérateur.

- $\mathcal{O}(n)$      $\mathcal{O}(n^2)$      $\mathcal{O}(\log n)$   
  $\mathcal{O}(1)$      $\mathcal{O}(n \log n)$

**Question 16** Le test d'équivalence (operator ==) pour un arbre binaire de recherche **non équilibré** a une complexité temporelle de \_\_\_\_\_. Supposez la meilleure implémentation possible de cet opérateur.

- $\mathcal{O}(n^2)$      $\mathcal{O}(\log n)$      $\mathcal{O}(1)$   
  $\mathcal{O}(n \log n)$      $\mathcal{O}(n)$

**Question 17 ♣** On ajoute, dans un ordre arbitraire,  $n$  entiers différents dans 2 arbres, l'un AVL et l'autre rouge-noir. Les 2 arbres ont une hauteur de  $h_1$  (AVL) et  $h_2$  (R-N). On peut conclure avec certitude :

- $h_2 \leq 1.5 \log_2 n + 1$      $h_2 \leq h_1$   
  $n < 2^{h_1}$      $h_1 \leq h_2$   
  $h_1 \leq 1.5 \log_2 n + 1$   
  $h_1 \leq 2 \log_2 n + 1$   
  $h_2 \leq 2 \log_2 n + 1$      $h_2 \geq \lceil \log_2 n \rceil$

**Question 18** La meilleure implémentation possible d'une copie profonde d'un arbre AVL (constructeur par copie et operator=) a une complexité temporelle de \_\_\_\_\_ dans le pire cas.

- $\mathcal{O}(\log n)$      $\mathcal{O}(n \log n)$      $\mathcal{O}(n)$      $\mathcal{O}(n^2)$

**Question 19** La meilleure implémentation possible d'une copie profonde d'un arbre rouge-noir (constructeur par copie et operator=) a une complexité temporelle de \_\_\_\_\_ dans le pire cas.

- $\mathcal{O}(n)$      $\mathcal{O}(n \log n)$      $\mathcal{O}(1)$      $\mathcal{O}(\log n)$      $\mathcal{O}(n^2)$

**Question 20** L'insertion dans un arbre AVL a une complexité temporelle de \_\_\_\_\_ dans le pire cas.

- $\mathcal{O}(n^2)$      $\mathcal{O}(n)$   
  $\mathcal{O}(n \log n)$      $\mathcal{O}(\log n)$   
  $\mathcal{O}(1)$

**Question 21** La meilleure implémentation possible de l'enlèvement dans un arbre binaire de recherche (**pas forcément équilibré**) a une complexité temporelle de \_\_\_\_\_ dans le pire cas.

- $\mathcal{O}(\log n)$      $\mathcal{O}(n^2)$   
  $\mathcal{O}(n \log n)$      $\mathcal{O}(n)$      $\mathcal{O}(1)$

**Question 22** Durant l'insertion d'un élément dans un arbre AVL, combien de rotation(s) peut-on avoir dans le pire cas? Une double rotation compte pour 2 rotations. Supposez que l'arbre contient  $n$  éléments.

- $2n$      $n$      $2$      $1$   
  $2 \log n$      $\log n$

**Question 23** Durant l'enlèvement d'un élément dans un arbre AVL, combien de rotation(s) peut-on avoir dans le pire cas? Une double rotation compte pour 2 rotations. Supposez que l'arbre a une hauteur de  $h$ . Choisissez la réponse la plus proche.

- $h$      $2h$      $1$      $2^h$   
  $2^{(h-1)}$      $2$

**Question 24** On insère les entiers 0 à 4 inclusivement dans un arbre AVL. L'ordre est aléatoire. Combien d'arbres différents (structure) peut on obtenir ?

- 8     $\geq 9$      $\leq 3$   
 6    5    7    4

**Question 25** On insère les entiers 0 à 63 (incl.), dans un ordre aléatoire, dans un arbre AVL. Quel est le plus petit entier pouvant se retrouver à la racine ?

- 11    13     $\leq 4$   
 12    5    9    6  
  $\geq 14$     7    10    8

**Question 26** On insère les entiers 0 à 63 (incl.), dans un ordre aléatoire, dans un arbre R-N. Quel est le plus petit entier pouvant se retrouver à la racine ?

- 11    10     $\geq 14$   
  $\leq 4$     5    8    7  
 12    13    6    9

**Question 27 ♣** Dans quel ordre doit-on insérer les nombres 1 à 3 dans un arbre AVL pour provoquer une double-rotation ?

- 3 1 2    1 3 2    2 3 1  
 1 2 3    2 1 3    3 2 1

**Question 28 ♣** Dans quel ordre doit-on insérer les nombres 1 à 7 dans un arbre AVL pour maximiser la hauteur de cet arbre ?

- 4 2 5 1 3 6 7    2 1 4 5 3 6 7  
 1 2 3 4 5 6 7    7 6 5 4 3 2 1  
 2 4 3 5 7 1 6    3 5 2 1 6 4 7

**Question 29 ♣** Dans un arbre rouge-noir, \_\_\_\_\_ (avant et après les opérations d'insertion et d'enlèvement): ...

- la racine est rouge.  
 les sentinelles sont à profondeur noire égale.  
 les feuilles sont à profondeur noire égale.  
 la hauteur = la profondeur noire des sentinelles.

**Question 30** Quelle est la profondeur noire des sentinelles après l'insertion des entiers 1 à 5 inclusivement dans un arbre Rouge-Noir?

- 3    1    4    0  
 5    2