Devoir 1

Vous devez remettre vos solutions sur Moodle **avant le vendredi 4 novembre 2022 à 23h55 sous la forme d'un unique fichier pdf**. Un retard de 24 heures au maximum sera accepté : pénalité de $\frac{m}{144}$ points, où m est le nombre de minutes de retard. **La note 0 sera attribuée au-delà d'un retard de 24 heures.** Le nombre total de points pour ce devoir est 100. Le devoir peut être fait en équipes de deux étudiant-e-s au maximum. Il doit être **intégralement** rédigé avec LaTeX.

Exercice 1 (**20 points**) On définit la fonction suivante :

fonction DF(Tab_1, Tab_2) **retourne** entier

- entrée : deux tableaux d'entiers Tab_1 et Tab_2 indicés de 1 à n
- sortie : un nombre entier x

```
début
    x ← 0                              c1
    pour i ← 1 haut n faire            c2
        pour j ← i haut n faire        c3
            si Tab1[i] = Tab2[j] faire c4
                x ← x + 1              c5
            fin si
        fin pour
    fin pour
    retourner x                        c6
fin DF
```

I. Calculer la valeur de DF(Tab_1, Tab_2) pour les tableaux suivants :

$Tab_1 =$

1	3	4	9	1	3	2	7

$Tab_2 =$

1	2	9	5	3	1	3	2

II. Quels tableaux correspondent au pire cas de cet algorithme ?
Quelle est alors la valeur de DF(Tab_1, Tab_2) ?

III. Donnez un exemple de tableaux correspondant au meilleur cas de cet algorithme ?
Quelle est alors la valeur de DF(Tab_1, Tab_2) ?

IV. Déterminer la complexité temporelle exacte de cet algorithme en fonction des temps d'exécution c_1, c_2, c_3, c_4, c_5 et c_6 :

a) Au pire cas,

b) Au meilleur cas.

Exercice 2 (**12 points**) Déterminer la somme suivante en utilisant les formules de sommations rappelées dans le chapitre 2 :

$$\sum_{i=33}^{n+2} \sum_{j=30}^{i+1} \sum_{k=15}^{j} 24i \text{ pour tout } n \in \mathbb{N}, n \geq 31.$$

Exercice 3 (**12 points**) On suppose que $f(n) = O\left(g(n)\right)$. Démontrer **à l'aide des définitions formelles** que $2\left(f(n)\right)^2 + 3\left(g(n)\right)^2 = O\left(\left(g(n)\right)^2\right)$.

Exercice 4 (**15 points**) Démontrer **à l'aide des définitions formelles** les égalités suivantes :

a) $n^3 = \omega\left(100n\right)$

b) $5n^3 \lg\left(10^n\right) = \Theta\left(\sqrt{1024^{\log_4(n)}}\ \frac{n^2}{\sqrt{n}}\right)$

Exercice 5 (**12 points**)
On considère l'équation de récurrence $T(n) = 6T(n-1) - 9T(n-2)$ pour tout entier $n \geq 3$, $T(1) = 3$, $T(2) = 18$.

a) Calculer $T(3)$, $T(4)$ et $T(5)$ à l'aide de l'équation de récurrence.

b) Démontrer **par récurrence** que $T(n) = n3^n$ pour tout entier $n \geq 1$.

Exercice 6 (**16 points**)
Utiliser, **si possible**, la méthode générale pour résoudre les récurrences suivantes :

a) $T(n) = 27T\left(\lceil \frac{n}{9} \rceil\right) + 5n^3$

b) $T(n) = 2T\left(\lceil \frac{n}{4} \rceil\right) + \sqrt{n}\lg^3(n)$

c) $T(n) = 20T\left(\lceil \frac{n}{10} \rceil\right) + n\lg(n)$

Exercice 7 (**8 points**) Pour chacune des formules suivantes, trouver la fonction **la plus simple possible** qu'on peut mettre à l'intérieur des parenthèses pour que l'égalité soit vérifée (**aucune justification requise**) :

a)
$$\frac{5n\lg\left(n^3\right)}{9\lg^3(n) + \sqrt{n}\lg(n)} = \Theta(\qquad)$$

b)
$$9^{2\log_3(n)} + 8^{3\log_4(n)} = \Theta(\qquad)$$

Exercice 8 (**5 points**) On considère un algorithme dont la complexité temporelle vérifie l'équation de récurrence $T(n) = 5T\left(\lceil \frac{n}{b} \rceil\right) + f(n)$.

Déterminer la valeur de b si le nombre de feuilles de l'arbre récursif de cet algortihme pour $n = 4096$ est égal à 15625. **Il ne faut pas dessiner l'arbre récursif** !