Université de Versailles – Saint-Quentin-en-Yvelines
Université Paris Saclay

le 23 octobre 2024

Master 1 Informatique 2024–2025
Compléments d'algorithmique et complexité
Examen

NOM : _____	Prénom : _____	Num. Étu. : 2☐☐☐☐☐☐☐☐

Aucun document autorisé. Aucun appareil électronique autorisé. Le barème est indicatif.

Exercice 1 – Complexité *(5 points)* **A remplir sur l'énoncé**

1. Si $f(n) = 10n + 100$, quelles expressions sont vraies :

   ☐ $f(n) = O(n)$ ☐ $f(n) = \Theta(n^2)$
   ☐ $f(n) = O(n \log n)$ ☐ $f(n) = \Omega(\log n)$

2. Classez en ordre croissant les quatre fonctions ci-dessous selon leur comportement asymptotique. Quelle est la plus petite fonction selon cet ordre ?

   ☐ $\frac{n^2}{\log_2 n}$ ☐ $\frac{n}{10^2}$ ☐ $31n^2 - 2n$ ☐ $2 \times \frac{\log_2 n}{2}$

3. Quelle est la complexité dans le pire cas d'un algorithme qui a $k + n$ étapes élémentaires avec $k < n$?

   ☐ $O(k \times n)$ ☐ $O(n)$ ☐ $O(n^k)$ ☐ $O(\log_2(k + n))$

4. Si le tri par permutation s'exécute en 1 seconde pour un tableau de taille 1000, en combien de temps s'exécutera-t-il pour un tableau de taille 10 000 ?

   ☐ 1000 secondes ☐ 100 secondes ☐ 1 seconde ☐ 10 secondes

5. Si l'on empile les lettres du mot `VOITURE` dans cet ordre (la première lettre empilée est le `F`, la dernière est le `E`) dans une pile, parmi les mots suivants, lequel n'est pas possible à obtenir grâce à des opérations *dépiler* éventuellement intercalées entre les opérations *empiler* ?

   ☐ `ERUTIOV` ☐ `OVTIRUE` ☐ `IOVTERU` ☐ `EVIOUTR`

Exercice 2 – ABR, Tas, AVL *(4 points)*

Soit la liste L des valeurs suivantes : $6, 11, 26, 28, 2, 3$. En partant d'une structure vide, vous devez dessiner l'arbre obtenu après l'insertion de chaque valeur de la liste L dans le cas :

1. d'une structure qui est un ABR ;
2. d'une structure qui est un tas minimum ;
3. d'une structure qui est un AVL.

Exercice 3 – Récursivité *(4 points)*

Soit un tableau T contenant n entiers. Les indices du tableau sont compris entre 0 et $n-1$. Chacune des valeurs du tableau est un compteur qui peut prendre l'une des valeurs suivantes :$\{0, 1, 2\}$. Pour incrémenter l'un des compteurs on utilise algorithme suivant :

```
1 Algorithme : Increment(T : tableau de n entiers, i : entier)
2 début
3 |  T[i] ← T[i] + 1
4 |  si T[i] = 3 alors
5 |  |  T[i] ← 0
6 |  |  Increment(T,(i+1) modulo n)
7 |  |  Increment(T,(i-1) modulo n)
8 |  fin
9 fin
```

1. Supposons que $n = 4$ et que $T = [2, 2, 2, 2]$. Que contient T après l'appel à **Increment(T,2)** ? Vous détaillerez votre réponse.
2. Qu'est-ce qui garantit dans les appels récursifs que le programme finira par s'arrêter ?

Tourner la page pour la suite des exercices.

Exercice 4 – Arbres Rouge-Noir *(7 points)*

Les arbres rouge et noir sont des arbres binaires de recherche. Dans ce modèle chaque noeud comporte un champ supplémentaire en plus de sa valeur qui est sa couleur. La couleur d'un noeud peut être soit *noire* soit *rouge*. Les feuilles de l'arbre ne contiennent aucune valeur. Tous les noeuds qui contiennent une valeur sont des noeuds internes de l'arbre.

Un arbre binaire de recherche est un arbre *rouge-noir* s'il satisfait les propriétés suivantes :

— Chaque noeud est soit *rouge*, soit *noir*.
— La racine est noire.
— Le père d'un noeud rouge est noir.
— Toutes les feuilles sont noires (mais ne contiennent aucune valeur).
— Tous les chemins issus d'un même noeud et se terminant en une feuille (descendante du noeud d'origine) ont le même nombre de noeuds noirs.

1. Est-il possible de colorier tous les noeuds de l'arbre binaire de recherche de la figure 1 pour en faire un arbre rouge-noir ? Justifier votre réponse. Les noeuds notés `F` sur la figure sont les feuilles.

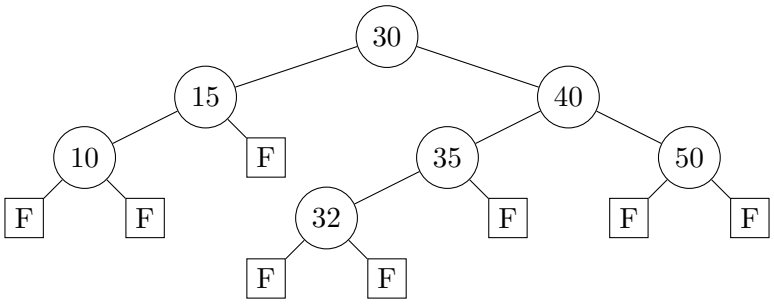

FIGURE 1 – Est-il possible de colorier cet arbre pour qu'il soit un arbre rouge-noir ?

2. Pour chaque arbre de la figure 2, dire s'il s'agit d'un arbre rouge noir. Si non, pourquoi ?
3. La structure de données pour un noeud d'un arbre rouge-noir est :

```
Enregistrement Noeud {
   Val     : entier;
   Couleur   : entier;
   Gauche : ↑ Noeud;
   Droit : ↑ Noeud;
}
```

Pour simplifier on supposera que toutes les valeurs stockées dans l'arbre sont positives ou nulles. Les feuilles auront pour valeur -1. La couleur rouge sera notée 1 dans le champ couleur et la couleur noire sera notée 2.

On définit la hauteur noire d'un noeud x, notée $H_n(x)$, comme le nombre maximum de noeuds noirs dans un chemin descendant de x (inclus) vers une feuille de l'arbre. La hauteur d'un arbre est le maximum parmi les hauteurs de tous les noeuds dans cet arbre. Ecrire un algorithme récursif qui calcule la hauteur noire d'un arbre enraciné en x.

4. Ecrire un algorithme récursif qui compte le nombre de noeuds ayant au moins une feuille comme enfant ?

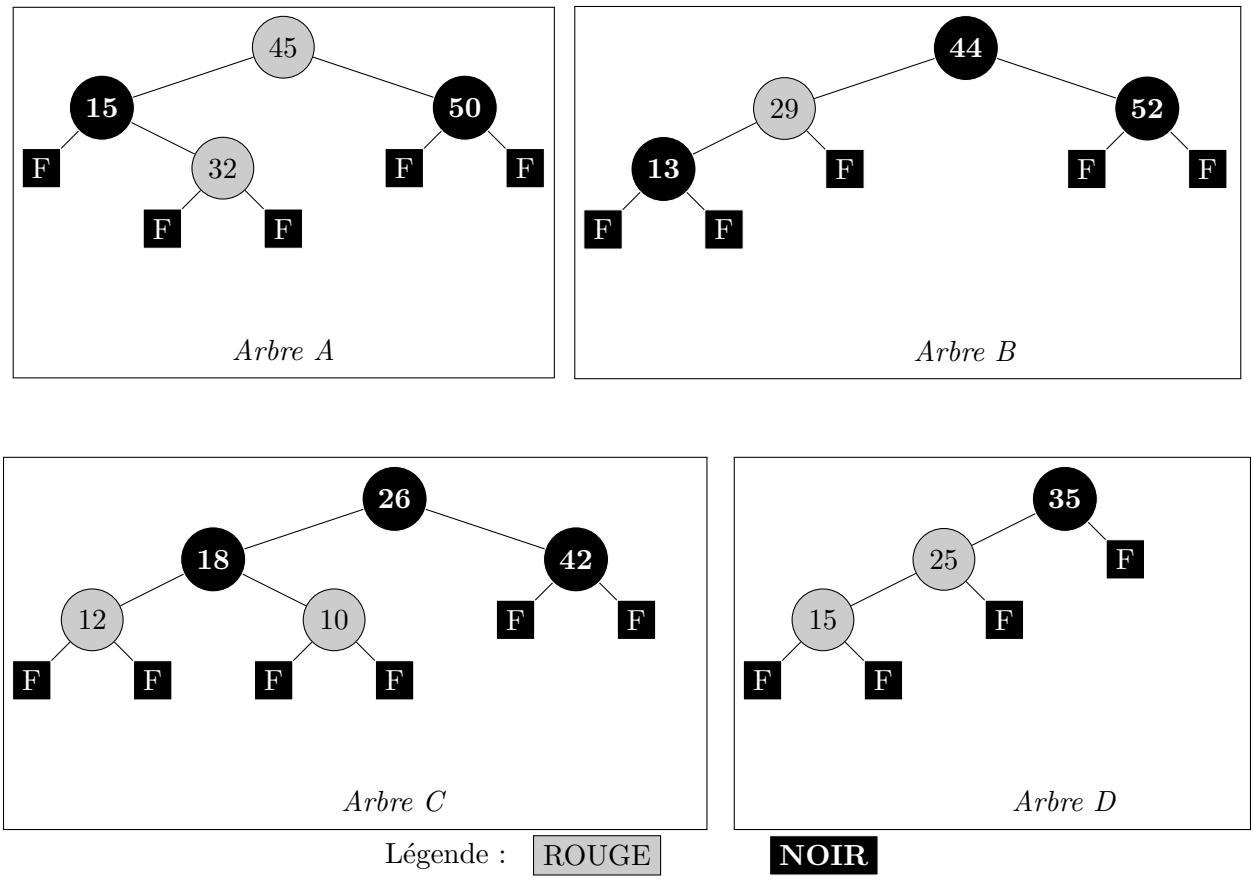

FIGURE 2 – Arbre rouge-noir ?