

## Examen du lundi 1er septembre 2008

Aucun document autorisé. Le barème est uniquement indicatif.

Vous pouvez écrire les algorithmes en C ou en pseudo-code.

### 1 Notation asymptotique, complexité

**5 points**

**Exercice 1.**

En rappelant les définitions, démontrer chacune des assertions suivantes.

3 pt  
18 min

1.  $\sum_{i=2}^n \log i = \Omega(n - 1)$
2. Si  $f = \Theta(h)$  et  $g = O(h)$  alors  $f + g = O(h)$
3. Il est faux de dire que, en général, si  $f = O(g)$  et  $g = \Omega(h)$  alors  $f = \Omega(h)$  (Donner un contre-exemple).

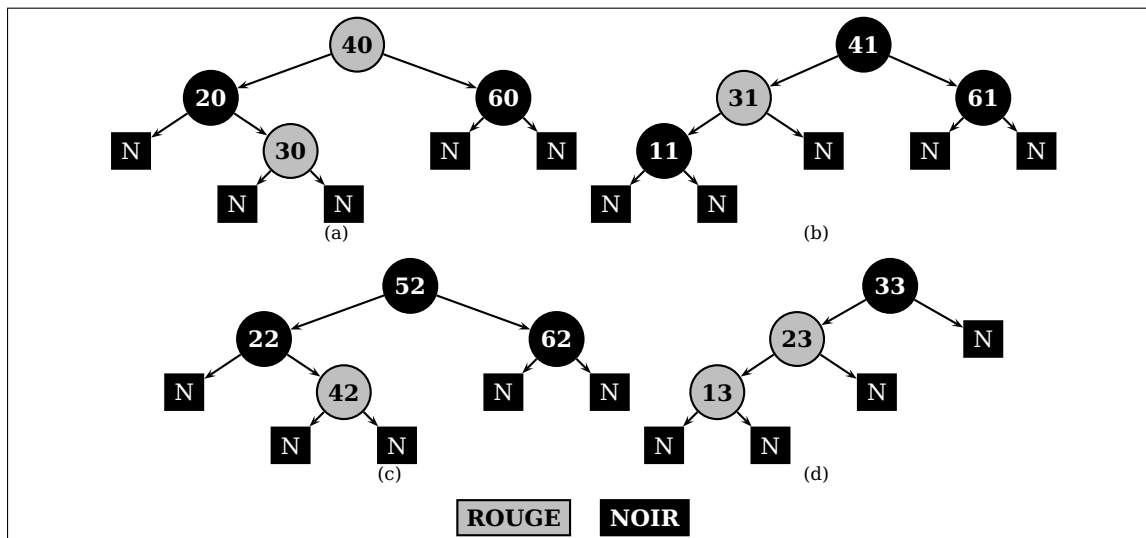
**Exercice 2.**

Classer les fonctions de complexité  $n \log n, 2^n, \log n, n^2, n$  par ordre croissant et pour chacune d'elle donner l'exemple d'un algorithme (du cours ou des TD) qui a asymptotiquement cette complexité en pire cas, en temps. Répondre dans un tableau en donnant le nom de chaque algorithme.

2 pt  
12 min

### 2 Structures de données

**6 points**



**Fig. 1:** Rouge noir ?

**Exercice 3.**

Pour chaque arbre de la figure 1, dire s'il s'agit d'un arbre rouge noir. Si non, pourquoi ?

1 pt  
6 min

**Exercice 4.**

Soit un tableau  $t$  contenant les éléments 10, 7, 3, 9, 11, 5, 6, 4, 8.

1. Former un tas max en insérant un à un et dans l'ordre les éléments de  $t$ . Répondre en représentant le tas obtenu. Supprimer l'élément maximum. Répondre en représentant le nouveau tas.
2. Dans le cours, il y a deux façons de former un tas max à partir d'un tableau d'éléments. On peut insérer un à un les éléments du tableau dans un tas (initialement vide) comme à la première question. Quelle est l'autre façon et quel tas obtient on pour l'entrée  $t$ ? Décrire l'algorithme et donner le tas obtenu.

1,5 pt  
9 min

2 pt  
12 min

**Exercice 5** (Insertion / suppression ABR).

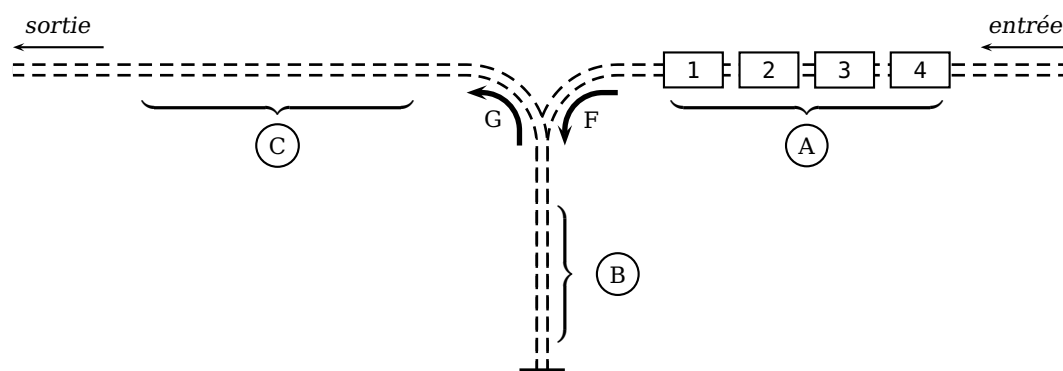
Former un arbre binaire de recherche en insérant successivement et dans cet ordre les éléments : 10, 7, 3, 9, 11, 5, 6, 4, 8 (répondre en représentant l'arbre obtenu). Supprimer l'élément 7. (répondre en représentant le nouvel arbre. Il y a deux réponses correctes possibles, selon la variante choisie pour l'algorithme de suppression).

1,5 pt  
9 min**3 Algorithmes - problèmes****9 points****Exercice 6.**

Écrire une fonction  $\text{Hauteur}(x)$  prenant un arbre binaire et rendant sa hauteur, c'est à dire le nombre d'éléments contenus dans la plus longue branche. Vous pouvez faire appel à des fonctions  $\text{Parent}(x)$ ,  $\text{Gauche}(x)$ ,  $\text{Droite}(x)$ .

1 pt  
6 min**Exercice 7.**

Soit le réseau ferroviaire suivant, représenté ici avec 4 wagons dans la zone  $A$  numérotés de 1 à 4.



Quatre actions sont possibles :  $\text{Ajouter}(\text{wagon})$  fait entrer un nouveau wagon sur le réseau (le wagon arrive par l'entrée dans la zone  $A$ ),  $\text{Retirer}()$  fait sortir un wagon (le wagon sort de la zone  $C$  par la sortie),  $F()$  fait passer un wagon de la zone  $A$  à la zone  $B$  et  $G()$  fait passer un wagon de la zone  $B$  à la zone  $C$ .

Par exemple, après ajout des wagons 1, 2, 3, 4 (comme sur l'illustration), la séquence :  $F()$ ,  $F()$ ,  $F()$ ,  $G()$ ,  $G()$ ,  $G()$ ,  $F()$ ,  $G()$ , suivie de quatre  $\text{Retirer}()$ , donnera l'ordre de sortie : 3, 2, 1, 4.

1. Si la séquence de wagons ajoutés est 1, 2, 3, 4, 5 dans cet ordre, peut-on obtenir en sortie les wagons dans l'ordre 3, 2, 5, 4, 1 ?
2. Donner une séquence de wagons en entrée, la plus courte possible, et un ordre de sortie que l'on ne peut pas réaliser.
3. On peut modéliser le réseau comme l'assemblage de trois structures de données vues en cours, une par zone (zone  $A$ , zone  $B$ , zone  $C$ ). Quelles sont ces structures de données ? Écrire (coder) les quatre actions du réseau à partir des opérations de base de ces structures de données.

1 pt  
6 min1 pt  
6 min1,5 pt  
9 min**Exercice 8.**

Étant donné un tableau  $T$  de  $N$  entiers et un entier  $x$ , on veut déterminer s'il existe deux éléments de  $T$  dont la somme est égale à  $x$ .

4,5 pt  
27 min

1. Donner un algorithme le plus simple possible, basé sur la comparaison, sans faire appel à des algorithmes du cours. Quel est le pire cas ? Donner un équivalent asymptotique du nombre de comparaisons dans le pire cas. (Justifier)
2. Pouvez-vous donner un algorithme en  $O(N \log N)$  comparaisons en pire cas ? (Justifier) Vous pouvez utiliser des algorithmes vus en cours et les résultats de complexité sur ces algorithmes.

