ORDRE DES INGÉNIEURS DU QUÉBEC

SESSION DE MAI 2017

Toute documentation permise Calculatrices : modèles autorisés seulement Durée de l'examen : 3 heures

14-LO-A1 -- Algorithmes et structures de données

QUESTION 1 [20 POINTS]

Analyse de complexité

a) [**9 points**] Pour chacun des algorithmes suivants, donner l'ordre asymptotique en utilisant la notation grand O.

```
Algo1(n)
  z = 0
  pour  x = 1 à n
     pour  y = 1 à x
        z = z + y
  pour j = 1 à n
     s = 1
     tant que s < n
        s = s * 4
```

```
Algo2(n,m)
  i = 1
  j = 1
  tant que i<=m et j<=n
     i = i+1
     j = j+1
```

```
Algo3(n)
  i = 1
  r = n
  tant que i<=1000
     r = r*n
     i = i+1
  retourner r
```

b) [**3 points**] Considérez l'algorithme suivant :

```
Algo(T[1..N])
  pour x ← 1 à N
     i ← 1
     tant que i < n
        y ← uniforme(0,1)
        T[i] ← T[i]*(1-y)+(y*x)
        i ← i*10
```

Notez que la fonction uniforme s'exécute en O(1). Supposons que l'implémentation de cet l'algorithme s'exécute en 12.3 secondes pour N = 10 000. Quel sera, approximativement, le temps d'exécution si N = 100 000?

c) [**4 points**] Considérez l'algorithme suivant :

```
FonctionMystere(T[1..n],a,b)
    si n == 1
        si T[1] > a et T[1] < b
            retourner T[1]
        sinon
            retourner 0

    n1 = FonctionMystere(T[1..n/3],a,b)
    n2 = FonctionMystere(T[n/3+1..2n/3],a,b)
    n3 = FonctionMystere(T[2n/3+1..n],a,b)
    retourner n1+n2+n3
```

i. Que fait cet algorithme?
ii. Donnez l'équation de récurrence de l'algorithme
iii. Utilisez le théorème général pour calculer l'ordre asymptotique de l'algorithme.

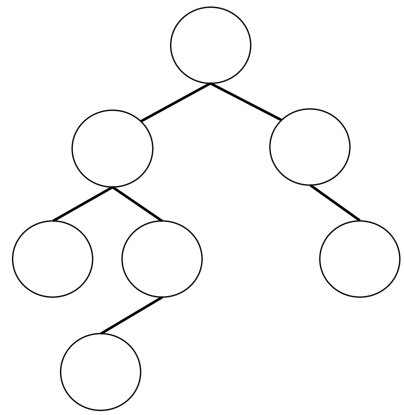
d) [**4 points**] Résoudre les équations de récurrence suivantes à l'aide du théorème général (*master theorem*) :

i. $T(n) = 8T(n/3) + n^2$
ii. $T(n) = 10T(n/2) + n^3$

QUESTION 2 [20 POINTS]

Arbres rouge-noir

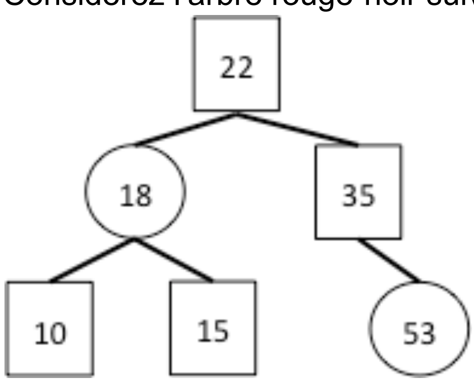
a) [**4 points**] Considérez l'arbre binaire suivant :



i. Placez les éléments 6,22,9,14,13,1,8 de telle sorte que les propriétés de l'arbre de recherche soient respectées.

ii. Étiquetez chaque nœud avec 'n' ou 'r' afin d'identifier les nœuds noirs et les nœuds rouges de telle sorte que l'arbre rouge-noir soit valide.

b) Considérez l'arbre rouge-noir suivant :



Notez que les carrés représentent les nœuds noirs et les cercles, les nœuds rouges.

i. [**4 points**] Insérez 65 dans l'arbre rouge-noir. Montrez chacune des étapes.

ii. [**4 points**] Insérez 40 dans l'arbre obtenu à la question (i). Vous devez avoir obtenu la bonne réponse à la question (i) pour avoir vos points à cette question. Montrez chacune des étapes.

iii. [**4 points**] Insérez 50 dans l'arbre obtenu à la question (ii). Vous devez avoir obtenu la bonne réponse à la question (ii) pour avoir vos points à cette question. Montrez chacune des étapes.

iv. [**4 points**] Insérez 38 dans l'arbre obtenu à la question (iii). Vous devez avoir obtenu la bonne réponse à la question (iii) pour avoir vos points à cette question.

QUESTION 3 [20 POINTS]

Heap

a) [**14 points**] À partir d'une structure vide, ajoutez successivement les éléments 21, 8, 3, 15, 17, 4 et 6 dans un min-heap. Montrez votre heap après chaque insertion.

b) [**6 points**] Dessinez tous les max-heap possibles contenant les éléments 2,4,6,8.

QUESTION 4 [20 POINTS]

Programmation dynamique

Considérez un environnement discrétisé de dimension 5 x 5. Associez à cet environnement, vous avez une fonction r(i,j) qui retourne la récompense obtenue si vous passez par la case (i,j) où i est le numéro de la ligne et j le numéro de la colonne. La fonction est définie comme suit :

i \ j	**1**	**2**	**3**	**4**	**5**
5	8	9	5	3	4
4	7	6	4	5	3
3	5	2	6	3	5
2	5	7	4	8	1
1	3	1	5	1	2

Vous pouvez démarrer votre quête à partir de n'importe quelle case de la première rangée (i=1). Vous pouvez vous déplacer d'une seule case à la fois, et ce, vers l'avant, diagonale avant droite et diagonale avant gauche. Donc, si vous êtes à la case (1,3), vous pouvez vous déplacer à la case (2,2), (2,3) ou (2,4).

Le problème est de trouver le chemin qui permet d'accumuler la plus grande récompense en démarrant à la rangée 1 et en terminant à la rangée 5.

On définit q(i,j) comme étant la récompense accumulée optimale lorsque vous êtes à case (i,j).

a) [**8 points**] Remplissez la matrice q(i,j).

	1	**2**	**3**	**4**	**5**
5					
4					
3					
2					
1					

b) [**8 points**] Donnez l'équation de récurrence

c) [**4 points**] Indiquez la récompense optimale et le chemin pour l'obtenir.

QUESTION 5 [20 POINTS]

Algorithme Glouton

a) [**12 points**] Examiner la balance de la Figure 1 sur laquelle on doit placer différents poids. Le problème consiste à placer les poids $q_o p_o$, $q_1 p_1$, … $q_{n-1} p_{n-1}$ sur la balance de manière à minimiser la différence de pesanteur (ou si vous préférez, de manière à maximiser l'équilibre de la balance). Écrivez un algorithme glouton effectuant cette tâche. Vous pouvez supposer que la description des poids (quantité et poids en grammes) se trouve dans deux tableaux. Le premier Q contient la quantité, le second, P, le poids en grammes.

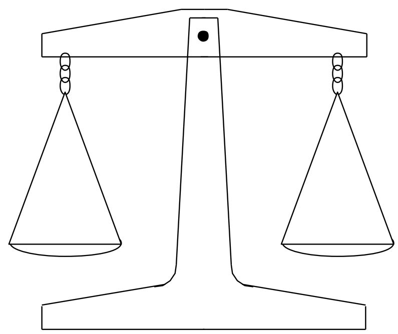


Figure 1

b) [**5 points**] Votre algorithme est-il optimal? Expliquer.

c) [**3 points**] Considérer maintenant la situation présentée dans la table I. En suivant la logique de votre algorithme, donnez la liste des poids présents sur chaque plateau de la balance.

libellé	poids	quantité
p_{30}	30	5
p_{15}	15	4
p_7	7	3
p_3	3	5

Table I