ORDRE DES INGÉNIEURS DU QUÉBEC

NOVEMBER 2017 SESSION

Open-book examination
Calculators : only authorized models
Duration : 3 hours

14-LO-A1 -- Algorithms and Data Structures

QUESTION 1 [20 POINTS]

Complexity analysis

a) [**12 points**] Consider the following algorithms

```
HasDouble1(T[1..n])
    for i = 1 to n
        for j = i to n
            if T[i] == T[j]
                return true
    return false
```

```
HasDouble2(T[1..n])
    QuickSort(T)
    for i = 1 to n-1
        if T[i] == T[i+1]
            return true
    return false
```

```
HasDouble3(T[1..n])
    HashTable ht
    ht.add(T[1])
    for i = 2 to n
        if T[i] is in ht
            return true
        else
            ht.add(T[i])
    return false
```

These algorithms determine if a value appears twice (or more) in the given array T. In the case the algorithm 3, (with the hashtable), consider that elements are uniformly distributed and that the table is big enough for the purpose of the algorithm.

i) For each algorithm, give the asymptotic growth in the best case scenario (Ω)
ii) For each algorithm, give the asymptotic growth in the worst case scenario (big O)

b) [**4 points**] Consider the following algorithm:

```
Mystery(T[1..n],a,b)
    if n <= 0
        return false
    if n = 1
        if T[1] > a et T[1] < b
            return true
        else
            return false
    if T[n/2] > a et T(n/2) < b
        return true
    if T[n/2] < a
        return Mystery(T[n/2+1..n],a,b)
    else
        return Mystery(T[1..n/2-1],a,b)
```

i. What is this algorithm doing?
ii. Give the recurrence equation.
iii. Use the master theorem to determine the asymptotic order.

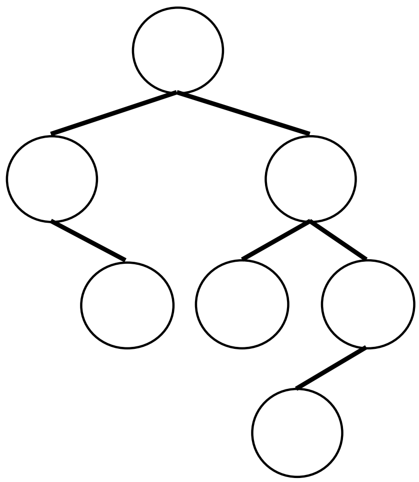
c) [**4 points**] Resolve the following equations using the master theorem:

i. $T(n) = 2T(n/4) + n^{0.51}$
ii. $T(n) = 3T(n/2) + n$

QUESTION 2 [20 POINTS]

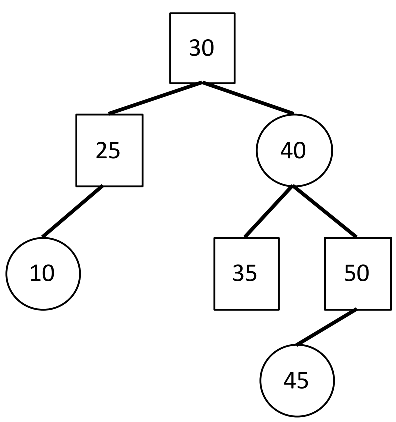
Red-Black Tree

a) [**5 points**] Consider the following binary tree:



i. Place the keys 8,22,17,1,14,30,40 in the tree so that the properties of a binary search tree are respected.

ii. Label each node with ‘b’ or ‘r’ to mark black and red nodes in order to obtain a valid red-black tree. Consider the following red-black tree:



a)

Note that black nodes are represented by a square and red nodes by a circle

b) [**5 points**] Insert 28 and 5 in the red-black tree. Show all your steps.

c) [**5 points**] Remove 30 and 10 from the **original** red-black tree. Show all your steps.

d) [**5 points**] From the original tree, insert two new keys in order to create a case 3. Show all your steps.

QUESTION 3 [20 POINTS]

Heap

a) [**16 points**] From an empty structure, insert the following elements (in the given order) in a max-heap: 3, 15, 18, 14, 16, 19, 22, 17. Show your heap after each insertion.

b) [**4 points**] Draw all possible min-heaps that contains the values 2,4,6,8

QUESTION 4 [20 POINTS]

Dynamic Programming

Consider the problem of counting the number of distinct binary strings of N bits without consecutive 1's. For example, for N=2, there are 3 possible strings: '00', '01','10'. For N=3, there are 5 possible strings: '000','001',010','100','101'

This problem can be solved using dynamic programming.

a) [**6 points**] Fill the following matrice M[1..2,1..N] for N = 10

	1	2	3	4	5	6	7	8	9	10
Chaîne se terminant par '0'										
Chaîne se terminant par '1'										

b) [**6 points**] Give the recursive relation for this problem
c) **5 points**] Give an algorithm to solve this problem for any value of N
d) [**3 points**] What is the order of growth of your algorithm and what is the solution for N=10?

QUESTION 5 [20 POINTS]

Greedy Algorithms

Consider the balance of Figure 1 on which different weights can be placed on two plates. The problem to solve consists of placing a set of weights $w_0.w_1,\ldots,w_N$ on both plates of the balance so that the weight difference between both plates is minimized.

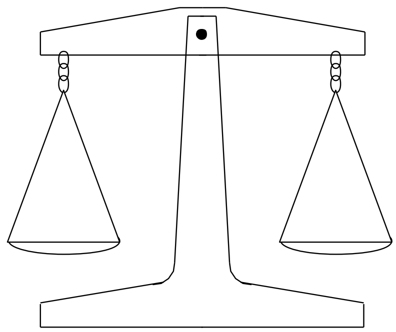


Figure 1

a) [**12 points**] Write a greedy algorithm to solve this problem.
b) [**5 points**] Is your algorithm optimal? Explain
c) [**3 points**] Use your algorithm to solve the problem using the weights of the following table:

Name	Weight	Quantity
w_{30}	30	5
w_{15}	15	4
w_7	7	3
w_3	3	5

Table I