Name :____

Place number :____

Code permanent :______________________________________

Directives :

- Write you name, place number and permanent code.
- Put your student card in view
- Read all questions and **answer directly on the questionnaire**.
- You can only use a pen or pencil, **no documentation, calculator, phone, computer or object**.
- This exam contains 10 questions for 175 points, including 10 bonus points.
- Approximately, we estimate about 1 point per minute.
- This exam contains 18 pages, including 3 detachable pages at the end for drafting.
- For developing questions, **write clearly and detail your answers**.
- You have 165 minutes to complete this exam.

GOOD LUCK!

1	/ 15
2	/ 15
3	/ 30
4	/ 10
5	/ 20
6	/ 15
7	/ 15
8	/ 15
9	/30
10	/10
Total	/165

1. (15) A crazy king has *n* bottles of wine and a spy poison one of them without telling which. A single drop even diluted in a ratio 1:1000000 causes death 1 month after ingestion. Develop a strategy to determine which bottle of wine has been poison in only one month and using in the O(log *n*) wine tasters.

2. (15) Show how to use a stack `S` and a queue `Q` to generate all the possible subsets of a set of *n* elements `T` in a non recursive manner.

3. (30) Two trees, T1 and T2, are said isomorphic if and only if one of the two following properties is satisfied: i) T1 and T2 are empty; or, ii) the roots of T1 and T2 possess the same number of sub-trees and the ith sub-tree of T1 is isomorphic to the ith sub-tree of T2 for i = 1..k, where k is the maximum number of children per node. The height of the smallest tree is *h*.

a) (15) Describe an algorithm (in python or pseudo-code) that tests if two trees are isomorphic.

b) (5) What is the execution time of your algorithm in the best case?

c) (10) What is the execution time of your algorithm in the worst case?

4. (10) Consider a priority queue implemented by using a min-heap in an array with indexes starting at 0.

a) (1) A which indexes(s) can be stored the smallest key?

b) (2) A which indexes(s) can be stored the third smallest key?

c) (3) A which indexes(s) can be stored the fourth smallest key?

d) (4) A which indexes(s) can be stored the ith smallest key? if $i > 1$?

5. (20) A group of kids wants to play a game called *Inmonopoly*, where at each turn the player with the biggest amount of money must give half of it to the player with the smallest amount of money. Which data structure(s) shall one use to play this game efficiently? Why?

6. (15) Consider the AVL search trees.

   a) (10) Insert the keys {1, 2, 3, 4, 5, 6, 7} in this order in an AVL tree that is initially empty. Draw the resulting trees after each insertion operation.

b) (5) Delete one by one the keys inserted in your AVL tree (from (a)) in increasing order. Draw the resulting trees after each delete operation. When a key must be replaced, use the predecessor. Note that if the last tree you obtained in (a) is wrong, you will not get any point here in (b).

7. (15) Consider the Splay trees.

a) (10) Insert the following keys {7,6,5,4,3,2,1} in this order in a Splay tree that is initially empty. Draw the resulting trees after each insertion operation.

b) (5) Access one by one the keys you inserted in the Splay tree in (a) in decreasing order. Draw the resulting trees after each access operation. Note that if the Splay tree you obtained in (a) is wrong, then you will not get any point here in (b).

8. (15) Imagine and draw a 2-4 search tree that has four different representations as red-black search trees and draw these four red-black search trees.

9. (30) Consider the Map ADT and the data structures Skip List and red-black search trees to implement it.

a) (1) How many pointers are needed to define a node in a Skip List?

b) (1) How many pointers are needed to define a node in a red-black search tree?

c) (10) What is the expected memory usage to store a Map of *n* keys by using a Skip List?

d) (8) What is the expected memory usage to store a Map of *n* keys by using a red-black search tree?

e) (10) Enumerate all good reasons that you know to implement a Map by using a red-black search tree rather than by using a Skip List.

10. (10) Consider the binary search trees. Insert the keys {3,14,15,92,65,35} in this order in a binary search tree initially empty. Draw the resulting trees after each insertion operation.

Draft:

Draft:

Draft: