EXAM RULES:

- This is a take-home exam, which means that you can use all and any study materials you need.
- Please complete this exam by yourself. It is expected that you will work by yourself and will not ask neither receive help for the exam from any other student or individual.
- Prepare your answers on separate sheet(s) of paper. You can either write your answers down by hand or prepare them on a computer and then print and submit that.
- **All exams must be submitted in hard-copy by Friday December 1 2017 21:00.**

GOOD LUCK!

Q1: Describe an augmenting data structure based on the binary search tree (BST) that can keep track of the height of left and right sub-tree of each BST node. Describe what additional information needs to be stored in each node of BST and how such information should be updated upon addition and removal of the nodes, to achieve that goal. What do you think such a data structure may be useful for?

Q2: Consider two sorted arrays A and B of length n_A and n_B (known in advance). Write down an algorithm for finding the median (that is, $(n_A+n_B)/2$-order statistic) of these two arrays in the worst-case time $\theta(n_A+n_B)$. If you cannot come up with a worst-case complexity $\theta(n_A+n_B)$ algorithm, then at least propose an algorithm that has expected-time complexity $\theta(n_A+n_B)$.

Q3: Consider an array of integers A. Write down a **dynamic programming** algorithm for finding the set of all integers S that can be represented as sums of numbers from A, without repetition. For example, for A={1,3,7} output S={1,3,4=1+3,7,8=7+1,10=7+3,11=7+3+1}

Q4: Consider the rod-cutting problem discussed in the class. That is, for a steel rod of length L that needs to be sold as a set of pieces ι_i, whereas each piece of length ι is valued as in the table below, find the largest possible value the rod can be sold for. Write down a **greedy algorithm** for this problem and apply that algorithm to construct a greedy solution for the rod of length L=9m.

The table of prices is for rod-cutting problem:

ι	1m	2m	3m	4m	5m	6m	7m	8m	9m	10m
price	1u	5u	8u	9u	10u	17u	17u	20u	24u	30u

Q5: In the class, during the discussion of the interval trees (IT), a following loop invariant was proposed to argue that IT-search algorithm was correct:

"Suppose that there is an interval I in IT that contains point t. Then during each iteration of the while loop, x is either nil OR the subtree rooted at x contains t."

Prove this loop invariant, thus proving that the IT-search algorithm is correct.

```
Algorithm IT-search(T,t):
x=T.root
while x≠nil and t is not in x
    if x.left≠nil and x.left.tmax≥t
        x=x.left
    else
      x=x.right
return x
```