Cheat sheet - (Official!!)

- both sides of one sheet of paper
- A4 paper size
- your name and group on it
- you can write anything as long as is **written by your own hand**
- at the end of the exam, the **cheat sheet** will be also delivered

Please respect conditions stated here;
otherwise, you will not be allowed with the cheat sheet.

No other additional resources!

What kind of subjects ?

Please see any question/problem/exercise discussed/presented on slides, lecture or seminar classes.

The final exam subjects will not consist of complex problems

- no problems as project problems

Complexity analisys

Ex: Consider Dynamic Vector:

a) D.S.
b) specify and design addLast, reserve
c) inserting a series of elements into a vector is a ***linear or quadratic*** time operation? Justify!

ADTs

- **(Short) ADTs**
  Stack, Queue, Forward Iterator, …
- **(Short) ADTs with some restrictions**
  ADT Forward Iterator.
  - use operations: hasNext, next
    next – move to the next element and return it
- **Part of some ADTs**
  ex.: modifiers operations

DS

- **representation (for a given ADT)**

Operations / subalg.

- **Specification & pseudocode**
- **Design (& with some restrictions)**
  Consider insertBefore operation for a singly linked list.
  Can the operation be done with list traversal, without or both?
  (Specify consequences & pseudocode for possible operations).

- **Give subalg. for similar with the studied subalg.**
  adapt studied algorithms
  some algorithms were only only discussed
  specify and design heapSort subalg. Present used d.s.

- **Use the subalg.**

Give (non-trivial) examples
Illustrate how subalg. works

Give 3 different degenerated BST containing values 1, 2, 3, 4
Insert a given values into a given RB-tree
describe the insert cases to be applied; show the result

True/false questions

Example:

Subject nr. 1

1. ADT Queue

2. Define a forward iterator for a binary tree; get elements on levels, left to right. (Do not use recursive subalgorithms)

a) present traversal idea (ex.: original subalg.+ how to split code)
b) representation & pseudocode;
specify any used or implemented operations
c) Print all the elements stored in a binary tree; use the iterator.
- specification & pseudocode
- illustrate/describe what happens if the tree is empty

3. Open addressing and double hashing.
Suppose we have a hash table of size 13, and:

$h_1(k) = k \bmod 13$

$h_2(k) = 1 + (k \bmod 11)$.

Illustrate the insertion of values 10, 22, 31, 4, 15, 17, 18, 19 into an initially empty hash table. (Present intermediary computations.)

4. An (almost) complete tree with height h has:
a) 2^h nodes b) 2^{h+1} nodes
c) between 2^h and 2^{h+1} nodes
d) between 2^h and 2^{h+1}-1 nodes