Data Structures and Algorithms I — Fri 3.2.2017

Exercise 2

No mandatory X-task yet this week, but there will be on third week, do tasks **??** and **??** to prepare for X-tasks.

1. Place the following functions in order of growth speed. Just one/few values of n is not enough. Which of the functions does not belong to group?

$\sqrt[3]{n}$	$42\text{loglog}n$	$n/\text{log}n$	$(\text{log}n)^2$
$\sqrt{n}\times n$	n^2-n^3	2^n	n
$n^{2.5}$	n^3	$n\text{log}n$	$\sqrt{n}-\text{log}n$
$n^{\log_2 7}$	2014	$\log(n^2)$	$\sqrt{n}/\log^2 n$
$n!$	$(3/2)^n$	n^2+n	

   Prepare to reason the order of functions.

2. Calculate the time complexity of the following program fragments with respect to n:

```
// A
int sum = 0;
for (int i = 1; i < 2*n; i++)
        sum = sum+1;

// C
int sum = 1;
while (sum < n)
        sum = sum + 2;

// E
int sum = 1;
while (sum < n)
        sum = sum + sum;
```

```
// B
int sum = 0;
for (int i = 0; i < n; i++)
        for (int j = 1; j < i; j++)
                sum = sum + 1;

// D
int sum = 0;
for (int i = −2; i < 2*n+3; i++)
        for (int j = 2; j < n−5; j++)
                sum = sum + 1;

// F
int sum = 0;
for (int j = 1; j <= n; j++)
        for (int i = 1; i <= j; i++)
                if (i == j)
                        for (int k = 0; k < n; k++)
                                sum = sum + 1;
```

3. Proof the following claims eiher true of false using the definition of O. I.e., deduct inequalities so that bounding from above will be shown for chosen c and n_0.

   (a) $n^2+2n+1=O(n^2)$

   (b) $n+n\sqrt{n}-9=O(n^2)$

   (c) $n+2n\sqrt{n}+3=O(n\sqrt{n})$

   (d) $n^3-4n^2+2n=O(n^3)$

4. Proof the following claims eiher true of false using the definition of O. I.e., deduct inequalities so that bounding from above or/and below will be shown for chosen c and n_0.

   (a) $3n^2-n+2=\Omega(n^2)$

(b) $n^2-n+2=\Omega(n^2)$

(c) $n\sqrt{n}-n+3=o(n^2)$

(d) $n^2-2n+2=\Theta(n^2)$

In the following "write an algorithm which" tasks, you should make a functional Java method that gets parameter input and possibly returns a new collection in accordance with the assignment, but does nothing else. Thus, for example, does not alter the input data (unless requested to do so) or print anything (at least in the final version). Please take the input generating main program from course web page. For exercise classes, we'll show your answers using projector, thus bring it on by saving it to the cs.uef.fi server, somewhere else in the network, or memory stick.

5. Write a linear time algorithm that gets as parameters two array-based lists of ascending order of elements (*ArrayList A*, *ArrayList B*) and which removes from list A all those elements that are present in list B. The algorithm returns the number of removed elements. Do not use ready *removeAll*() -operaation or set types. State the time complexity of your algorithm.

6. Write a linear time algorithm that gets as parameters two linked lists of ascending order of elements (*LinkedList A*, *LinkedList B*) and which removes from list A all those elements that are present in list B. The algorithm returns the number of removed elements. Do not use ready *removeAll*() -operaation or set types. State the time complexity of your algorithm.