Introduction to algorithms, flowcharts, pseudocode of an algorithm.

Recursive algorithm - one that invokes (makes reference to) itself repeatedly until a certain condition matches, which is a method common to functional programming.

Iterative algorithms use repetitive constructs like loops and sometimes additional data structures like stacks to solve the given problems.

Usefull links:

If you know some programming language (C, C++, Java, Pascal) you can use its syntax instead of described pseudocode syntax. Everything that would be understandable for me will be counted as a correct pseudocode.

Examples of some flowcharts:
- Algorithm which compute \( n! \): http://www.rff.com/n_factorial_flowchart.htm
- Euclid algorithm: http://motivate.maths.org/conferences/conf124/c124_Euclid.shtml

I recommend to you program „RAPTOR” where you can create flowchart and execute it to be sure that it is correct:
- Raptor’s webpage: http://raptor.martincarlisle.com/

Tasks for classes

Task 1
Draw a flowchart of an algorithms which checks whether you win in a lotto lottery. You have a lotto ticket with 6 numbers sorted ascending, and you have to compare them with a 6 numbers draw by a lotto machine (there are also given in ascending order). We are only interested whether we hit all six numbers or not.

I prepare a file with reading data in this problem, so you can use it http://www.staff.amu.edu.pl/~mw/DNIF/lotto.rap.

Task 2 Binary search is an algorithm for locating the position of an element in a sorted list by checking the middle, eliminating half of the list from consideration, and then performing the search on the remaining half. Write in pseudocode iterative and recursive version of this algorithm.

If you’re not familiar with this algorithm you can read http://www.articledashboard.com/Article/The-Binary-Search-Algorithm/1047768.

Task 3

**Task 4**


**Task 5**

Write a pseudocode of an iterative and recursive version of an algorithm that computes $n$-th element of a sequence defined by:

$$
F_n := \begin{cases} 
0 & \text{dla } n = 0; \\
1 & \text{dla } n = 1; \\
2 & \text{dla } n = 2; \\
F_{n-1} + F_{n-2} + F_{n-3} & \text{for } n > 2.
\end{cases}
$$

(so the beginning of a sequence is as follows 0,1,2,3,6,11,20,...). Please implement in RAPTOR the iterative version, check what is the 25-th element of this sequence.

**Additional information**

There would be some tasks in the final test to draw a flowchart and write pseudocode of a given algorithm (on paper). So there is no homework from this class.