**<https://github.com/andkris/DAA_20191002>**

**Algorithm Analysis and Design. Laboratory work.**

1. Optimize the code provided bellow using Dynamic Programing technique

**static** **long** FF(**long** n){

**long** k = 0;

**if** (n > 1){

k += 2 \* FF(n - 1);

k += 2 \* FF(n - 2);

} **else**{

k = 1;

}

**return** k;

}

1. Write the function of the complexity, which refers recursive equation:
   1. T(n) = 2T(n/2) + n
   2. T(n) = 2T(n-10) + n
   3. T(n) = T(n-1) + n

Which function is the “most expensive”? Find the maximum n, which can be used to solve equation in acceptable (let’s say 5 sec. using the current computer”).

1. Analyze the provided example for Rostering problem. Review, how the results are influenced by
   1. changing GA parameters (population size, mutation rate, etc.)
   2. changing constraints (i.e. min free days period, max. working time, etc.)
   3. changing coefficients describe the schedule goodness.
2. Modify example in such way, that objective function would take working time into account.
3. Check how changes results, if in “createScheduleAfterMutation” function uncommenting line with code “initialWorkerTasks.addAll(skipedTaskList)”. Explain the changes.

**Andrius Kriščiūnas,** [**andrius.krisciunas@ktu.lt**](mailto:andrius.krisciunas@ktu.lt)

Faculty of Informatics | Department of Applied Informatics

Kaunas University of Technology, Kaunas, Lithuania