Algorithms and Data Structures

Adigeev Mikhail Georgievich
mgadigeev@sfedu.ru
adimg@yandex.ru

The course lecturer

Name: Adigeev Mikhail Georgievich

E-mail: mgadigeev@sfedu.ru

adimg@yandex.ru

Course site at Moodle:

http://edu.mmcs.sfedu.ru/course/view.php?id=613

Course format

- Course = lectures + practical lessons
- Lectures: 1 lesson per week
- Practical lessons: 1 lesson per week
- 4 modules:
  - 6 programming assignments + 4 tests.
  - 60 points max
- Final exam (40 points max).
  - 38+ points in classes to be admitted
  - 22+ points in the exam to pass

Course structure

Module 1. *Introduction to algorithms and data structures*.

2 assignments + 1 test

Module 2. *Greedy algorithms. Divide-and-Conquer strategy*.

2 assignments + 1 test

Module 3. *Dynamic programming*.

1 assignment + 1 test

Module 4. *NP-hard problems*.

1 assignment + 1 test

Reference textbooks & useful resources

1. The course section at the Moodle: http://edu.mmcs.sfedu.ru/course/view.php?id=613
2. Open Data Structures. An open content textbook. URL:http://opendatastructures.org/
3. Jeff Erickson. Algorithms. A free electronic version. URL: http://jeffe.cs.illinois.edu/teaching/algorithms/
4. Clifford A. Shaffer. Data Structures & Algorithm Analysis. A free for educational use electronic book. URL: https://people.cs.vt.edu/~shaffer/Book/
5. The Computer Science Handbook. A Reference for Data Structures and Algorithms. URL: https://www.thecshandbook.com/
6. Lectures Notes on Algorithm Analysis and Computational Complexity (Fourth Edition) - Ian Parberry: URL: http://ianparberry.com/books/free/license.html
7. Kleinberg J., Tardos É. Algorithm Design. — Pearson Education Inc., 2006.
8. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd ed. — The MIT Press — 2009.

Programming assignments

- Programming language: C++ (or C)
- Program should have command–line interface
  - ✓GUI is admissible but does not influence the score
- Data input and output via text files
- A solution for a programming assignment should include a zip file with:
  - ✓ a ‘Project’ folder, containing all necessary source (and header) files
  - ✓ an executable file (Release, Win32)
  - ✓ sample input and output files
  - ✓ a .bat file for running the program with command line arguments

Command line interface

```
Program1.exe In.txt 10 Out.txt
```

```
int main(int argc, const char * argv[])
{
    if (argc == 4)
    {
        // argv[0] = program file name
        string InputFile = argv[1];
        int BufLen = atoi(argv[2]);
        string OutputFile = argv[3];
        ...
        return 0;
    }
    else
    {
        cout << "Invalid number of arguments: " << argc << " instead of 3." << endl;
        return 1;
    }
}
```