Combinatorial Optimization

MSc Jarosław Synak

1. The goal of this laboratory is to prepare a heuristic algorithms, which solves a given NP-hard problem.
2. Students can be absent two times without explanation, every next unexplained absence will lower the final grade.
3. Projects are prepared in pairs. Members of each pair can be from different laboratory groups.
4. Students are supposed to implement their solutions in C++ (it has to compile under g++ 6.0 compiler on *Linux*) or Python (any version), *make* can be used. **Programs have to read everything from *stdin* and print to *stdout*** (exact input and output formats will be given). **Any suspicious operations like accessing the filesystem, printing unnecessary things to *stdout*, using network connection can result in failing grade.** Remember, projects which do not follow the rules described above will be **automatically rejected**!
5. Algorithms have 5 min to print the solution, so it is advised to implement a timer. If after 5 min no solution is returned – **project will be rejected**. If you kill the program after 5 mins it can exit before printing all the output, it is highly recommended to flush the standard output beforehand.
6. All the information about the instance comes from the standard input, so tuning the parameters based on e.g. file name is strictly forbidden!
7. The solution has to be sent to me before the deadline (**2024-01-14**) with an instruction how to compile and run it (e.g. which Python version with which plugins was used).
8. Together with the algorithm, a complete description of the method used should be sent. Necessary parts:
   - Names of the authors and their student IDs, name of the subject, student group
   - Theoretical part, covering the method with a short explanation why it was chosen, what are the advantages.
   - Implementation part – a description how the algorithm was implemented
   - Conclusion and sources (literature and webpages used)
9. After sending your project via email, you also have to come personally (write an email before to let me know), so I can ask you about your solution and confirm that you understand what happens in your code.

10. Your grade will be based on how complex is your approach, quality of your description and how well you can answer my questions.
11. At the end of the semester, your solutions will be compared and the best one will be chosen.
12. Example approaches (can be combined):
    - Branch and bound
    - Genetic algorithm
    - Simulated annealing
    - Tabu search
    - Deep learning