Assignment 3: study the approximation of NP-hard problem solutions

Part 1: validation set

- Pick any relevant/interesting NP-hard problem and implement an exhaustive search solution. You are free to use any best-in-class tools available (e.g., SAT solvers) with an optimal solution guarantee.
- Establish a set of inputs for testing (can be random, but verifiable; for tough problems implement a solution checker) and a benchmarking infrastructure.
- Provide an estimation of the work required in Θ-notation.

Part 2: optimization & approximation

- Combine all the learned methods to develop an algorithm that solves the same problem from Part 1. Use approximation techniques with guaranteed deviation from the optimum.
- Provide reasoning, algorithm designing steps, and resulting complexity estimation.
- Observer and record the performance characteristics of the solution. Compare it to best-in-class optimal solvers.

Notes on submission

- Deadline: 11:59 pm Dec 20th
- Form: a single PDF per part and/or a link to a GitHub repo. Inline comments for complexity analysis.
- Where to send: petr.kurapov@gmail.com with email topic in the form:
  - <surname> computational complexity 2024 assignment 3