UNIVERSITÀ DI PISA
Dipartimento di Informatica
Corso di Laurea in Informatica

Assignment 03

Competitive Programming and Contests

Prof. Rossano Venturini

Giacomo Trapani - 600124

Academic Year 2024/2025

Exercise 1 This exercise requires the implementation of an algorithm to find out the maximum number of attractions which can be visited throughout the holidays given constraints on the duration of such holidays, the number of attractions per day and the time complexity of the algorithm which is expected to be $O(n \times D^2)$ with n the number of cities and D the number of days.

The solution given makes use of dynamic programming:

- **base case** $n = 1$: we can visit either D attractions or the number of attractions available in the city, whichever is the lowest.
- **n cities** given the solution for the previous $n - 1$ cities: for every $x \in [0; D]$, we can spend $D - x$ days in the optimal solution thus far and x days in the n-th city. When we are done, we compare against the previous optimal solution.

The dynamic programming table has size $n \times D$.

Exercise 2 This exercise requires the implementation of an algorithm to find out the maximum number of topics which can be explained throughout a course given constraints on their order in $O(n \times \log(n))$ with n the number of courses.

Considering every topic as a pair (beauty, difficulty), we sort the topics for their beaut, then we run a "longest increasing sequence" algorithm using their difficulty as key.

How to run The code can be run via `cargo run`. The following directory structure is expected

```
src
├── lib.rs
└── main.rs
tests-1
tests-2
Cargo.toml
Cargo.lock
```