**Learning objectives**

Upon completion of the course, I achieved the following goals:

***1. Algorithmic problem solving***

I've learned how to break down tasks into smaller, manageable components.

I have mastered the skills of designing and analyzing algorithms, which allowed me to solve problems faster and more efficiently.

***2. Data Structures***

Studied advanced data structures such as trees, graphs, and heaps.

I understood how and when to use these data structures in various competitive programming tasks.

***3. Dynamic programming***

I mastered dynamic programming techniques, which helped me solve complex problems that require optimization.

I practiced using these techniques to improve the performance of solutions.

***4. Greedy algorithms***

I understood the essence of greedy algorithms and their application.

I learned how to apply greedy strategies to solve optimization problems.

***5. Algorithms on graphs***

He studied the basic algorithms on graphs, such as BFS, DFS and Dijkstra's algorithm.

I solved graph-related problems, which greatly improved my graph skills.

***6. Mathematics in competitive programming***

He deepened his knowledge in the field of number theory, combinatorics and probability.

He applied these mathematical concepts to solve algorithmic problems.

***7. Programming skills***

Improved programming skills, especially in terms of ensuring readability, efficiency and correctness of the code.

He practiced implementing algorithms and data structures in a competitive environment.

***8. Strategies for participation in competitions***

He has developed effective strategies for participating in programming competitions.

I learned how to manage time and choose tasks during contests, which helped me improve my results.

**The list of tasks that have been solved:**

*Competitive problems:*

* Love Triangle
* Party
* Rook, Bishop and King
* Perfect Premutation
* Party\_2.0
* Card Exchange
* Rectangle Filling
* Everything Nim

**Projects:**

* Wildcards

This project implements a search in a string by a mask having "?" replacing 1 character and "\*" replacing as many characters as you want

* Huffman coding

A project that allows you to apply Huffman encoding to any text file.

* Friend (Social Network)

A class has been developed that has methods for creating a person in the database, removing a person from the database, adding friendship between people, and finding friends.

The project allows you to display a list of a person's friends up to two handshakes.

* Binary Tree

A classical binary tree, with output of sorted vertices. Whether the tree is binary and other.

* Sierpinski Fractal

Allows you to draw the Sierpinski triangle using user-defined points. From the depths of rendering, filling triangles, marking the grid, etc.