Map of Tasks

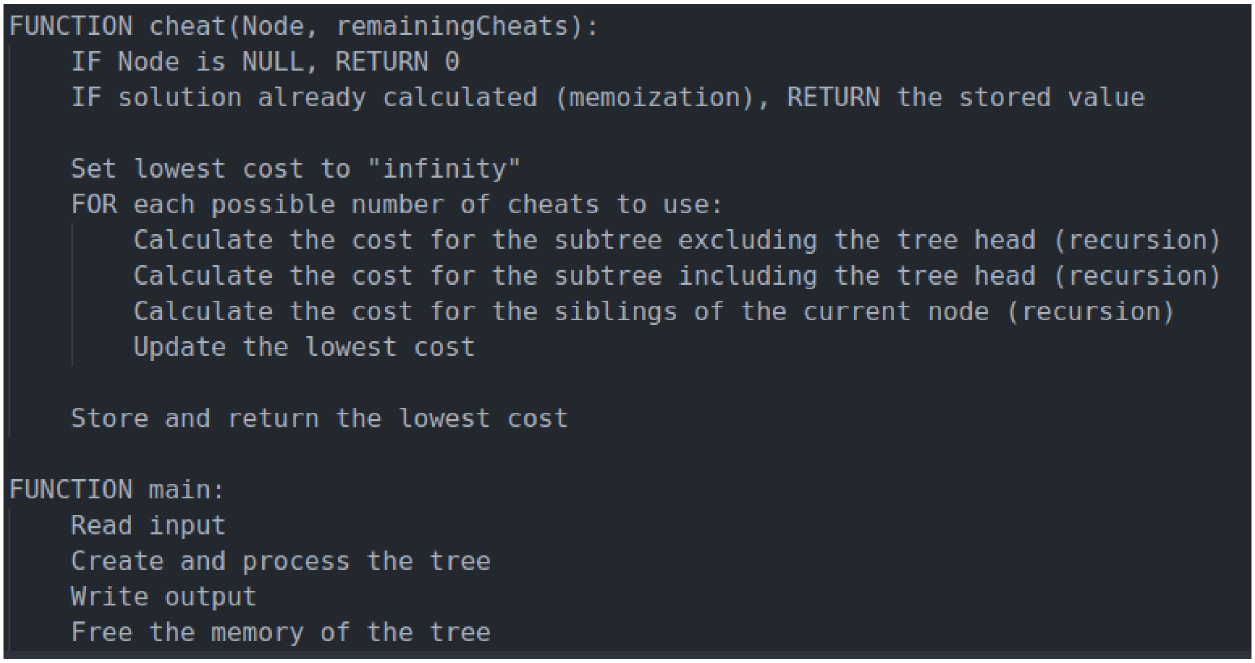
Problem description

The International Informatics Olympiads in Teams (IIOT) pose a unique challenge in the field of programming.

In this specific problem, the goal is to minimize the total time required to complete a set of assigned tasks.

Each task depends on another, forming a tree structure of tasks, with the possibility of using a limited number of "cheats" to reduce the completion time of a task.

Algorithm pseudocode

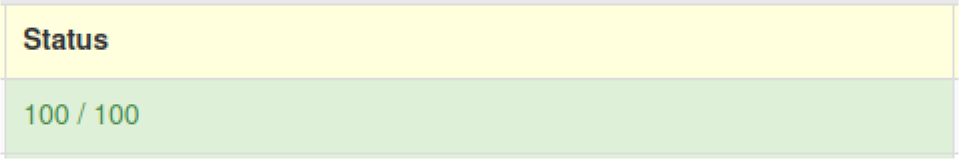


```
FUNCTION cheat(Node, remainingCheats):
    IF Node is NULL, RETURN 0
    IF solution already calculated (memoization), RETURN the stored value

    Set lowest cost to "infinity"
    FOR each possible number of cheats to use:
        Calculate the cost for the subtree excluding the tree head (recursion)
        Calculate the cost for the subtree including the tree head (recursion)
        Calculate the cost for the siblings of the current node (recursion)
        Update the lowest cost

    Store and return the lowest cost

FUNCTION main:
    Read input
    Create and process the tree
    Write output
    Free the memory of the tree
```

Time complexity analysis

- Creating the tree has a complexity of O(N), where N is the number of nodes.
- The `cheat` function has a time complexity of O(N*C) in normal cases, and $O(N*C^2)$ in the worst case: this is because, for each node, it examines all possible combinations of cheats up to the maximum limit (C), and each recursive call can involve up to C iterations.

Test



Status
100 / 100

Known issues

- **Scalability**: The algorithm may become less efficient with a very high number of tasks or cheats.
- **Optimization**: Some parts of the algorithm could be optimized for better performance.