

EXERCISE REPORT

TEAM 15

GEOMETRIC ALGORITHMS 3D

Lecturer: Son Nguyen Thanh MCS

Class: CS112.N21.KHTN

Member 1: Quan Vo Minh – 21520093

Member 2: Nhat Nguyen Viet – 21520378

Excercise: Team 6's Excercise

Date: 03/06/2023

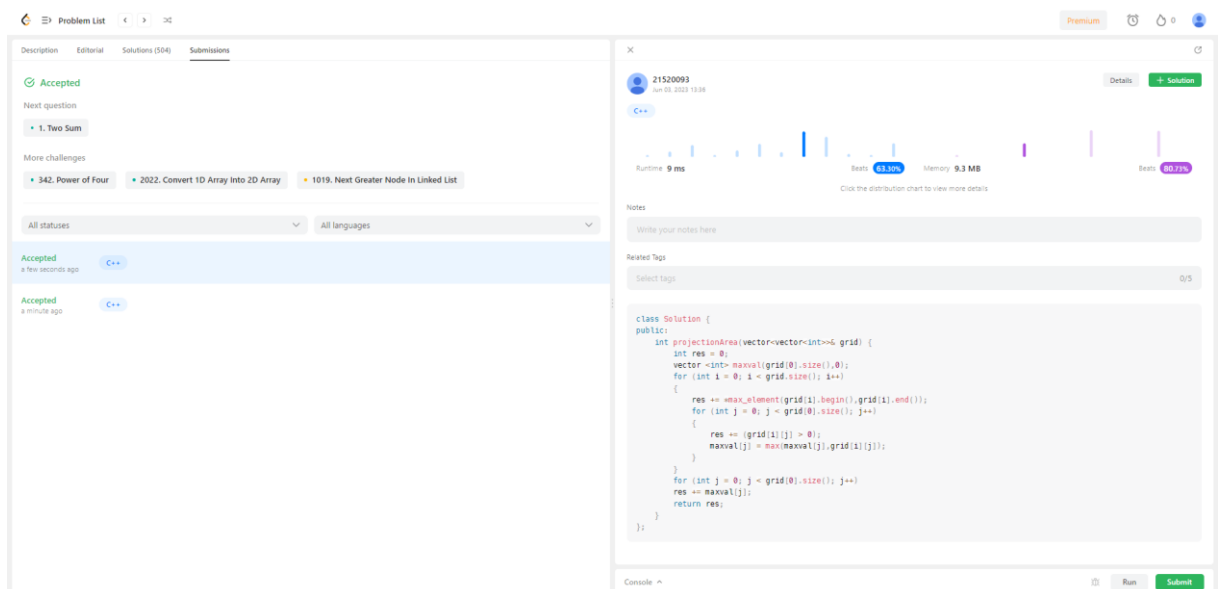
EXERCISE 1: Code convex hull 3D and submit in [SPOJ](#). Write a report and capture screen your submit AC in your report and write your solution

ID	DATE	USER	PROBLEM	RESULT	TIME	MEM	LANG
31512184	2023-06-19 15:30:42	omlgg	Convex Hull 3D	accepted edit ideone.it	0.26	5.4M	CPP14

Algorithm: Incremental algorithm

- Create tetrahedron (initial convexhull)
 - Pick 2 points p_1 and p_2
 - Pick a point not lying on the line p_1p_2 to create a plane $p_1p_2p_3$
 - Pick a point not lying on the aforementioned plane (if all points are on the same plane, use 2D convexhull algorithm)
- Randomize the remaining points P
- For each $p_i \in P$, add p_i into the convexhull
 - If p_i lies inside or on the boundary of convexhull then do nothing.
 - If p_i lies outside of convexhull insert p_i .

EXERCISE 2: Solve this problem in leetcode and capture your screen your submit AC in your report



The screenshot shows a LeetCode submission for problem 21520093. The left pane displays the problem description and submission status (Accepted). The right pane shows the C++ code for the solution, which calculates the maximum area of a projection of a 3D grid onto the xy-plane. The code uses nested loops to iterate over the grid and calculate the area of the projection.

```

class Solution {
public:
    int projectionArea(vector<vector<int>>& grid) {
        int res = 0;
        vector<int> maxval(grid[0].size(), 0);
        for (int i = 0; i < grid.size(); i++) {
            res += max_element(grid[i].begin(), grid[i].end());
            for (int j = 0; j < grid[0].size(); j++) {
                res += (grid[i][j] > 0);
                maxval[j] = max(maxval[j], grid[i][j]);
            }
        }
        for (int j = 0; j < grid[0].size(); j++) {
            res += maxval[j];
        }
        return res;
    }
};

```