Competitions
Overview
Copilot Opportunities
Design
Development
UI Development
QA and Maintenance
CloudSpokes
Algorithm
High School
The Digital Run
Submit & Review

TopCoder Networks
Events
Statistics
Tutorials
Forums
Surveys
My TopCoder
Help Center
About TopCoder

Member Search:
Handle: Go
Advanced Search

Dashboard > TopCoder Competitions > ... > Algorithm Problem Set Analysis > SRM 608

Search

TopCoder Competitions

SRM 608

View | Attachments (22) | Info

Browse Space

Added by vexorian , last edited by vexorian on Feb 14, 2014 (view change)
Labels: (None) EDIT

Archive
Match Overview
Discuss this match

Single Round Match 608
Friday, February 7th, 2014

Match summary

This problem set was brought to you by algorithm director **rng_58**. A tough match that put emphasis in mathematical thought. Division 1 coders began having to solve a problem that required some good analysis to reveal the greedy solution. **Petr** solved this problem in seemingly supernatural 4 minutes. The mathematical reasoning was not missing from the graph theory division 1 medium either, **Petr** also had a stunning speed in this problem solving it correctly in under 12 minutes. Many great coders solved the division 1 hard, but **WJMZBMR** did it in glorious 17 minutes. Some similar speed records (second fastest submission in medium, fourth fastest submission in easy), allowed **WJMZBMR** to score a significant division win over **Petr** (2nd place), **ainu7** (3rd), **tomek** (4th) and **mR.ilchii** (5th). Division 2 actually faced similar themes and complexities, with slightly simpler versions of the division 1 easy and medium. The fastest submission in their division 2 problem gave **andyooo** the division 2 win.

The Problems

OneDimensionalRobotEasy | MysticAndCandiesEasy | BigOEasy | MysticAndCandies | BigO | OneDimensionalRobot

BigO

Rate It | Discuss it

Used as: Division One - Level Two:

Value	600
Submission Rate	102 / 743 (13.73%)
Success Rate	59 / 102 (57.84%)
High Score	**Petr** for 516.31 points (11 mins 50 secs)
Average Score	305.32 (for 59 correct submissions)

The explanation for the division 2 version explains how to identify unbounded graphs (those in which we need to return -1) and also a proof that also finds a method to calculate the degree of the polynomial in bounded graphs.

Code

The specific division 1 code:

```
static const int MAX_N = 50;
int m;
bool compressed[MAX_N];
int degree[MAX_N];
int g[MAX_N][MAX_N];
int dp[MAX_N];

// finds the path with the maximum number of compressed nodes starting at
// node x in the SCC graph.
int rec(int x)
{
    int res = dp[x];
    if (res == -1) {
        res = 0;
        for (int i=0; i < degree[x]; i++) {
            res = std::max(res, rec(g[x][i]) );
        }
        if ( compressed[x] ) {
            res++;
        }
    }
    dp[x] = res;
    return res;
}

int minK(vector<string> graph)
{
    // Find the SCCs in the graph:
    int n = graph.size();
    int reach[n][n];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            reach[i][j] = ( (graph[i][j] == 'Y') ? 1 : 0 );
```

Alternative solutions and additional comments.

<Place your comments here>

Next problem: OneDimensionalRobot

By **vexorian**

TopCoder Member

Editorial feedback Part 1	Choose
I liked it.	
I didn't like it.	

Editorial feedback division 1 hard	Choose
I liked it.	
I didn't like it.	

3 Comments | Add Comment