

C6-J-这是一道凸包？

题目描述

题目描述

BellalaBella 刚刚学习了计算几何中的凸包知识，她想到了这样一个问题。

二维平面中有 n 个点，第 i 个点的坐标为 (x_i, y_i) ，权值为 w_i 。Bellalabella 可以选择一个点集 S ，设 m 和 M 分别为点集 S 中点的横坐标最小值和最大值，此方案的权值为 S 中所有点与 $(m, 0), (M, 0)$ 构成的凸包面积减去选取点的权值和。特别地，如果 $S = \emptyset$ 则权值为 0。

Bellalabella 想知道所有选取方案中最大权值为多少。

输入格式

第一行一个正整数 t ($1 \leq t \leq 10$)，表示数据组数。

对于每组数据，第一行一个正整数 n ($1 \leq n \leq 5 \times 10^3$)，含义同题目描述。

接下来 n 行，每行三个非负整数 x_i, y_i, w_i ($0 \leq x_i, y_i, w_i \leq 10^4$)，含义同题目描述。

可能存在两个不同的点坐标相同。

输出格式

对于每组数据，输出一行一个浮点数，保留一位小数，表示最大权值。

解题思路

对所有点按横坐标排序。

记 $dp[i]$ 为最右边点为 i 的最大权值。

对于每个 i ， $dp[i]$ 的初始值应为 i 的权值。

$dp[i]$ 的结果由 $dp[j]$ ($j < i$) 加上这两个点之间的梯形面积减去 i 的权值更新。

注意到点都是整点，则面积和的两倍一定是整数，为避免精度影响，在计算时所有面积和点权可以按两倍算。

输出结果时可用 `printf("%d.%d\n", ans / 2, ans % 2 ? 5 : 0);`。

即

$$dp[i] = \max_{j < i} dp[j] + (i_y + j_y) * (i_x - j_x) - 2 * i_w$$

最后输出 dp 的最大值即可

实现代码

```
#include <bits/stdc++.h>
using namespace std;

struct point {
```

```

struct point {
    int x, y, w;
    bool operator<(const point &p) const {
        return (x < p.x) || (x == p.x && y < p.y);
    }
};

point p[5003];
int dp[5003], n;

void solve() {
    scanf("%d", &n);
    for (int i = 0; i < n; i++) scanf("%d %d %d", &p[i].x, &p[i].y, &p[i].w);
    sort(p, p + n);
    int ans = 0;
    for (int i = 0; i < n; i++) {
        dp[i] = -2 * p[i].w;
        for (int j = 0; j < i; j++) {
            dp[i] = max(dp[i], dp[j] + (p[i].y + p[j].y) * (p[i].x - p[j].x) - 2 * p[i].w);
            ans = max(ans, dp[i]);
        }
    }
    printf("%.2d\n", ans / 2, ans % 2 ? 5 : 0);
}

int main() {
    int t;
    scanf("%d", &t);
    while (t--) solve();
    return 0;
}

```