

ABC 112 解説

drafear, square1001, E869120

2018 年 10 月 6 日

A: Programming Education

この問題では、入力形式が 2 通りあります。これにどのように対処すればよいでしょうか。実は、「条件分岐」で対処することができます。最初に 1 つの整数 N を入力して、もし $N = 1$ であれば「Hello World」と出力し、そうでないならば ($N = 2$ ならば)「残り 2 つの整数 A, B を入力して $A + B$ を出力する」とすればよいです。入力は、上から順 (同じ行の文字は左から順) に読み込まれるので、 $N = 2$ のときは $2 \rightarrow A \rightarrow B$ の順に読み込まれるから、上の方法が通用します。

【擬似コード】

```
input N
if N = 1:
    print "Hello World"
else:
    input A, B
    print A + B
```

【サンプルコード】

サンプルコード (C++) : <https://beta.atcoder.jp/contests/abc112/submissions/3340558>

サンプルコード (Java) : <https://beta.atcoder.jp/contests/abc112/submissions/3340568>

サンプルコード (C#) : <https://beta.atcoder.jp/contests/abc112/submissions/3340582>

サンプルコード (Python 3) : <https://beta.atcoder.jp/contests/abc112/submissions/3340592>

B: Time Limit Exceeded

$t_i \leq T$ を満たす i のうち c_i の最小値を求めつつ、そのような i が存在するか調べていきます。これを実現するアルゴリズムとして、例えば次のようなアルゴリズムが考えられます。

アルゴリズム 1

1. $ans := \infty$ とする
2. $i = 1, 2, \dots, N$ と順に、 $t_i \leq T$ かつ $c_i < ans$ なら $ans := c_i$ とする
3. $ans = \infty$ なら条件を満たす i は存在しないので **TLE** と出力し、そうでなければ ans を出力する

アルゴリズム 2

1. $t_i \leq T$ であるような i について c_i を集めた配列を c' とする
2. c' が空なら **TLE** と出力し、そうでなければ c' の最小値を求めて出力する

以下は、これらのアルゴリズムをそれぞれ C++ で実装した例です。

アルゴリズム 1

```
#include <iostream>

using namespace std;

const int inf{int(1e9)};

int main() {
    int N, T; cin >> N >> T;
    int ans{inf};
    for (int i = 0; i < N; ++i) {
        int c, t; cin >> c >> t;
        if (t < T && c < ans) {
            ans = c;
        }
    }
    if (ans == inf) {
        cout << "TLE" << endl;
    }
    else {
        cout << ans << endl;
    }
}
```

アルゴリズム 2

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int main() {
    int N, T; cin >> N >> T;
    vector<int> c(N), t(N);
    for (int i = 0; i < N; ++i) {
        cin >> c[i] >> t[i];
    }
    vector<int> filtered_c;
    for (int i = 0; i < N; ++i) {
        if (t[i] <= T) {
            filtered_c.push_back(c[i]);
        }
    }
    if (filtered_c.size() == 0) {
        cout << "TLE" << endl;
    }
    else {
        cout << *min_element(filtered_c.begin(), filtered_c.end()) << endl;
    }
}
```

C: Pyramid

さて、問題文の制約をよく見てみましょう。一番重要なのは、以下の二点です。

- 条件を満たすピラミッドは 1 つしか存在しない。
- ピラミッドの中心座標は (C_x, C_y) であり、 C_x, C_y は 0 以上 100 以下の整数。

まず、条件を満たすピラミッドがただ 1 つなのであれば、必ず以下のものが 1 個以上存在します。

- 調査で得られた情報のうち、 $h_i \geq 1$ を満たす情報 (x_i, y_i, h_i) 。

なぜなら、全部 $h_i = 0$ であるとき、調査を行っていない整数座標を中心とする高さ 1 のピラミッドが条件を満たしてしまうので、最低でも $101 \times 101 - N$ 個、 $N = 100$ の場合 10,101 個以上の条件を満たすピラミッドが存在することになるからです。そこで、前述の条件を満たす情報のうち 1 個を (x_t, y_t, h_t) とおきます。

C_x, C_y は 0 以上 100 以下の整数なので、中心座標として考えられるのは高々 $101 \times 101 = 10,201$ 通りです。そこで中心座標 (px, py) を全探索することを考えます。

px, py が決まっているとき、ピラミッドの高さは $h_t + |px - x_t| + |py - y_t| (h_t \geq 1)$ となるので、一意に定まります。その後は、その (中心座標, 高さ) の組が調査によって得られた情報に適合するか、を確認するだけです。

計算回数は $101 \times 101 \times N$ 回くらいになります。300 点問題にしては結構実装は重いです。

【サンプルコード (C++)】

<https://beta.atcoder.jp/contests/abc112/submissions/3340382>

D: Partition

a_1, a_2, \dots, a_N の最大公約数を D とすると、 a_1, a_2, \dots, a_N は D で割り切れるので $M = a_1 + a_2 + \dots + a_N$ も D で割り切れます。従って、 D は M の約数になります。

また、 D が M の約数のとき、 $a_1, a_2, \dots, a_N \geq D$ なので $M \geq N \times D$ です。

実は D が M の約数であって $M \geq N \times D$ を満たすとき、最大公約数が D 以上となるように a_1, a_2, \dots, a_N が構成できます。具体的には、 $a_1 = a_2 = \dots = a_{N-1} = D, a_N = M - (N-1) \times D$ とします。このとき、 a_N は正整数かつ D の倍数なので各要素の最大公約数は D 以上*1となります。

したがって、 M の約数かつ $\frac{M}{N}$ 以下の整数のなかで最大のが答えとなります。約数は (約数のペアの片方が平方根以下であることを利用して)、 $O(\sqrt{M})$ で求められることが知られているため、この問題は $O(\sqrt{M})$ で解くことができます。

*1 $N = 1$ のとき最大公約数は M となり D より大きくなることがあります