ABC 124 解説

writer: drafear

2019 年 4 月 13 日

A: Buttons

大きい方のボタンを選んで押すことを 2 回繰り返すのが最適です。基本的には同じボタンを押しますが、$A = B$ のときは両方のボタンを 1 回ずつ押すのが最適なので、注意する必要があります。

実装では、以下の 3 つのパターンについて場合分けすると比較的簡単に書くことができます。

- $A > B$ の場合: $A + (A - 1)$ が答えになります。
- $A < B$ の場合: $B + (B - 1)$ が答えになります。
- $A = B$ の場合: $A + B$ が答えになります。

C++ での実装例を以下に示します。

```cpp
#include <bits/stdc++.h>

using namespace std;

int main() {
  int A, B; cin >> A >> B;
  int ans;
  if (A > B) {
    ans = A + (A-1);
  }
  else if (B > A) {
    ans = B + (B-1);
  }
  else {
    ans = A + B;
  }
  cout << ans << endl;
}
```

B: Great Ocean View

入力を配列で受け取り、各 $i = 1, 2, ..., N$ について全ての $j = 1, 2, ..., i-1$ に対して $H_i \geq H_j$ かを確認し、そうであった i の数を数えることで答えを求めることができます。入力を配列に保存しておくことで後から自由に何度も使えるので、実装がしやすくなるかもしれません。C++ での実装例を以下に挙げます。

```cpp
#include <bits/stdc++.h>

using namespace std;

int main() {
  // input
  int N; cin >> N;
  vector<int> H(N);
  for (int i = 0; i < N; ++i) {
    cin >> H[i];
  }
  // compute
  int ans = 0;
  for (int i = 0; i < N; ++i) {
    bool is_candidate = true;
    for (int j = 0; j < i; ++j) {
      if (H[j] > H[i]) {
        is_candidate = false;
        break;
      }
    }
    if (is_candidate) {
      ++ans;
    }
  }
  // output
  cout << ans << endl;
}
```

C: Coloring Colorfully

1 箇所色を決めれば他の色が全て決まり、パターンとしては 白黒白黒・・・ または 黒白黒白・・・ しかありません。この 2 つのパターンとそれぞれ何箇所異なっているかを調べ、その最小値が答えになります。時間計算量は $O(|S|)$ です。

D: Handstand

指示を一度行うたびに、人の状態が切り替わる部分を 2 箇所まで潰すことができます。これを踏まえて、連続して 1 を並ばせる左端を i 番目としたときにいくつ 1 を並べることができるかを考えます。

- $S_i = 0$ のとき: i 番目以降で $S_j \neq S_{j+1}$ となる場所を順に最大 $2K$ 箇所潰すのが最適です。
- $S_i = 1$ のとき: i 番目以降で $S_j \neq S_{j+1}$ となる場所を順に最大 $2K+1$ 箇所潰すのが最適です。

1 を並べる左端を i としたときの最大値を X_i とすると、答えは $\max\{X_1, X_2, ..., X_N\}$ となります。しかし、これをナイーブに実装しても $O(N^2)$ となり間に合いません。

そこで、何かしら工夫する必要があります。まず、i (連続して 1 を並ばせる左端) として、元の文字列で 0 または 1 が連続する始点のみを探索すれば十分です。例えば、"11000..." の 4 文字目や 5 文字目の 0 を左端とするよりも 3 文字目の 0 を左端とした方が良いでしょう。このような場所を左から順に $1 = i_1 < i_2 < ... < i_r$ とします。また、便宜上 $k > r$ なる k について、$j_k = N+1$ とします。このとき、$k = 2, 3, ..., r$ について、$S_{i_k} \neq S_{i_k - 1}$ です。よって、$k = 1, 2, ..., r$ について、X_{i_k} は

- $S_{i_k} = 0$ のとき: $X_{i_k} = i_{k+2K} - i_k$
- $S_{i_k} = 1$ のとき: $X_{i_k} = i_{k+2K+1} - i_k$

となり、答えは $\max\{X_{i_1}, X_{i_2}, ..., X_{i_r}\}$ です。これは、$O(N)$ で動作します。

別解として、二分探索を使った $O(NlogN)$ の解法や、しゃくとり法を用いた $O(N)$ の解法もあります。