

ABC 103 解説

writer: drafear

2018年7月21日

A: Task Scheduling Problem

タスクを完了する順序を全通り調べて最小値を求める方法でも解けますが、実装が若干大変です。実は、 A_i の小さいタスクから順に、または A_i の大きいタスクから順に完了したときに最小となり、 A_1, A_2, A_3 の最大値と最小値の差が答えになります。

C++ 言語による解答例

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

using namespace std;

int main() {
    int A1, A2, A3; cin >> A1 >> A2 >> A3;
    int ans = max({A1, A2, A3}) - min({A1, A2, A3});
    cout << ans << endl;
}
```

B: String Rotation

$|S|$ 回操作を行うと元に戻るので、0 回以上 $|S| - 1$ 回以下の回数操作を行った後に T と一致するものがあるか調べれば十分です。

C++ 言語による解答例

```
#include <iostream>

using namespace std;

int main() {
    string S, T; cin >> S >> T;
    bool ans = false;
    for (int i = 0; i < S.size(); ++i) {
        if (S == T) {
            ans = true;
            break;
        }
        S = S.back() + S.substr(0, S.size()-1);
    }
    if (ans) {
        cout << "Yes" << endl;
    }
    else {
        cout << "No" << endl;
    }
}
```

C: Modulo Summation

$m = a_1 \times a_2 \times \dots \times a_N$ とすると、各 i について $m \bmod a_i = 0$ なので、 $(m - 1) \bmod a_i = a_i - 1$ です。したがって $f(m - 1) = (a_1 - 1) + (a_2 - 1) + \dots + (a_n - 1)$ となり、各項が最大値を取っているのが f の最大値です。実装では、 m は非常に大きいため、 m を計算せずに $f(m - 1)$ を直接計算すればよいです。よって、 $O(n)$ でこの問題が解けました。

D: Islands War

まず、 (a_i, b_i) の組を b_i について昇順にソートします。ソートした順に、以下を行います。

- 既にその要望を満たしている場合、すなわち、既に 1 本以上橋を取り除いていて、直前に取り除いた橋によってその要望が満たされている場合、何もしない。ここで、直前に取り除いた橋によってその要望が満たされている場合とは、西から x 番目の橋を直前に取り除いたとすると、 $a_i \leq x$ であることである。
- まだその要望を満たしていない場合、西から $b_i - 1$ 番目の橋、すなわち西から $b_i - 1$ 番目の島と b_i 番目の島を接続する橋を取り除く。

このとき、最適となります。なぜなら、後者の場合について、西から a_i 番目, $a_i + 1$ 番目, ..., $b_i - 1$ 番目の橋のうち 1 つ以上取り除かなければなりません。が、 $b_i - 1$ 番目より手前の橋 (西から x 番目の橋とする) を取り除いたとき、後に残る部分問題 (西から $x + 1$ 番目以降の島と橋のみについての問題) を考えれば必ず損するからです。したがって、ソートがボトルネックとなり、 $O(n \log n)$ で解けました。

余談: 実は $O(n)$ でも解くことができます。興味があれば考えてみてください。