ABC 170 解説

evima, gazelle, kort0n, tozangezan, ynymxiaolongbao, yuma000

2020 年 6 月 14 日

For International Readers: English editorial will be published in a few days.

A: Five Variables

$x_1, \ldots, x_5$ を標準入力から読み込んだ後は、if 文を 5 つ連ねて 5 つの変数のうちどれが 0 であるかを直接確かめることも、$15 - x_1 - x_2 - x_3 - x_4 - x_5$ が答えとなることを利用することもできます。

それぞれの方針における Python での実装例を示します。

```
X = list(map(int, input().split()))

if X[0] == 0:
    print(1)
elif X[1] == 0:
    print(2)
elif X[2] == 0:
    print(3)
elif X[3] == 0:
    print(4)
else:
    print(5)
```

```
print(15 - sum(list(map(int, input().split()))))
```

B: Crain and Turtle

(原案: yuma000, 準備・解説: evima)

方針 1

発言の前半「庭の動物の総数は X 匹」が正しいと仮定すると、考えられる鶴と亀の数の組合せは「鶴 0 匹、亀 X 匹」「鶴 1 匹、亀 $X-1$ 匹」...「鶴 X 匹、亀 0 匹」の $X+1$ 通りです。これらのそれぞれにおける足の総数を計算し、その結果が一度でも Y 本となれば答えは Yes、一度も Y 本とならなければ答えは No です。素直に実装するなら、以下の Python のコードのように for 文を使うことになるでしょう。

```python
X, Y = map(int, input().split())
ans = 'No'
for a in range(X + 1):
    b = X - a
    if 2 * a + 4 * b == Y:
        ans = 'Yes'
print(ans)
```

方針 2

答えが Yes であることの必要十分条件は、Y が $2X$ 以上 $4X$ 以下の偶数であることです。

必要性の証明: a 匹の鶴と b 匹の亀について発言が正しいとすると、$Y = 2a + 4b = 2(a + 2b)$ であり、a, b が整数であることから Y は偶数でなければなりません。また、$a + b = X, 2a + 4b = Y$ から $a = (4X - Y)/2, b = (Y - 2X)/2$ が得られ、a, b が非負であることから Y は $2X$ 以上 $4X$ 以下でなければなりません。よって示されました。

十分性の証明: Y が $2X$ 以上 $4X$ 以下の偶数であれば、$(4X - Y)/2, (Y - 2X)/2$ はともに非負整数となり、発言が正しいような鶴と亀の数の組合せとして「鶴 $(4X - Y)/2$ 匹、亀 $(Y - 2X)/2$ 匹」が存在します。よって示されました。

C: Forbidden List

(原案: tozangezan, 準備・解説: evima)

$X, p_1, \ldots, p_N$ はいずれも 1 以上 100 以下ですが、答えは 0 または 101 となりうる点に注意を要します。

この点にさえ注意すれば、与えられる値の小ささのため計算を高速化する工夫などは必要なく、「書かれていることを実装しなさい」という通常の Beginner Contest 問題 B 相当の問題です。本問が問題 B として出題されていたなら以下に具体的な実装方法を述べるところですが、問題 C を解かれる方にそのような説明は不要と考え、省略します。

D: Not Divisible

エラトステネスの篩と同様の処理を, 数列 A の要素についてのみ行います.
サイズ A_{max} の bool 配列 dp を用意し, true で初期化します.$dp[i]$ =true のとき, i より小さい i の約数が A に存在しないことを表します. 数列の要素を昇順に見ます.x が数列 A に含まれているとき, x より大きい x の倍数 y について, $dp[y]$ を false に変更します.
同じ値について処理を複数回行わないように実装すると, 調和級数を用いた解析により, 時間計算量が $O(A_{max} \log A_{max} + N \log N)$ であることが分かります.
同じ値が複数存在する場合に, それらを答えに含めないことに注意してください.

E. Smart Infants

ある幼稚園の中で最もレートの高い幼児を、その幼稚園の最強園児と呼ぶことにします。
以下のようなデータを管理することで時系列順に答えを求めてゆくことができます。

- それぞれの園児が所属する幼稚園の番号
- それぞれの幼稚園に所属する幼児のレートの順序付き多重集合（幼稚園一つに対して一つ）
- それぞれの幼稚園の最強園児のレートの順序付き多重集合（全体に対して一つ）

順序付き多重集合は C++ の multiset などを用いることで高速に処理することができます。
それぞれのクエリでは、以下のような変更操作を行います。

- 最強園児のレートの集合から、転園する園児の元の幼稚園の元の最強園児のレートを、削除する
- 最強園児のレートの集合に、転園する園児の元の幼稚園の新しい最強園児のレートを、挿入する（園児が一人もいない場合何もしない）
- 最強園児のレートの集合から、転園する園児の新しい幼稚園の元の最強園児のレートを、削除する（園児が一人もいない場合何もしない）
- 最強園児のレートの集合に、転園する園児の新しい幼稚園の新しい最強園児のレートを、挿入する
- 転園する園児の元の幼稚園のレートの集合から、転園する園児のレートを、削除する
- 転園する園児の新しい幼稚園のレートの集合に、転園する園児のレートを、挿入する
- 転園する園児の所属する幼稚園の番号を更新する

以上の変更操作の後、最強園児のレートの集合の中の最も小さい値を出力すれば良いです。計算量は $O(NlogN)$ です。

F: Pond Skater

Dijkstra 法を用います。以下のようなグラフを考えます。
次のような状態量をグラフの頂点と見做します。

- 現在いるマス
- すぬけ君が東西南北のどの方向を向いているか

次のように遷移します。

- 向いている方向に 1 つ進んだマスに蓮がないなら、「向いている方向に 1 進んだマスにいて、向きはそのまま」という状態量に、今の状態量までの最短コストに $1/K$ を足したコストで到達可能とする。
- 「同じマスにいて、向きが 90 度回転した」という状態量に、今の状態量までの最短コストの小数点以下を切り上げたコストで到達可能とする。

計算量は $O(HWlog(HW))$ です。