

AtCoder Beginner Contest 096 解説

問題作成者 : E869120, square1001

2018 年 5 月 5 日

For international readers: English editorial starts from page 5.

A – Day of Takahashi

1 月 1 日 から a 月 b 日までの間には, 少なくとも 1 月 1 日, 2 月 2 日, ..., $a - 1$ 月 $a - 1$ 日の, 合計 $a - 1$ 日の高橋があり, これはどのような (a, b) の組に対しても成り立ちます.

また, a 月 a 日も高橋であるので, $a \leq b$ であれば追加で 1 日を足せばよいです. よって, 答えの場合分けは以下のようになります.

- $a \leq b$ のとき, 高橋は a 日
- そうでないとき, 高橋は $a - 1$ 日

C++ や Java などのプログラミング言語では, if-else 文を用いて条件分岐をすることで, 場合分けをすることができます.

ソースコード (C++) : <https://beta.atcoder.jp/contests/abc096/submissions/2456270>

ソースコード (Java) : <https://beta.atcoder.jp/contests/abc096/submissions/2457128>

B – Maximum Sum

A, B, C は正の数なので、「一番大きい数を 2 倍することを K 回繰り返す」という方法が、最終的な $A + B + C$ の値を最大化する方法です。証明は以下ようになります。

- $K = 1$ の時は、数 p を 2 倍すると数の合計が p 増えるので明らか
- $K = 2$ の時は、一番大きな数を 2 倍してもこの数が 3 つの数の中で一番大きいのに変わりはないから、 $K = 1$ の時の状態に持っていける。残り 2 回のときの増分を最大化しながら残り 1 回の時の増分も最大化できるので、自明にこの方法が最適。
- $K \geq 3$ のときも同様に、帰納的に示せる。

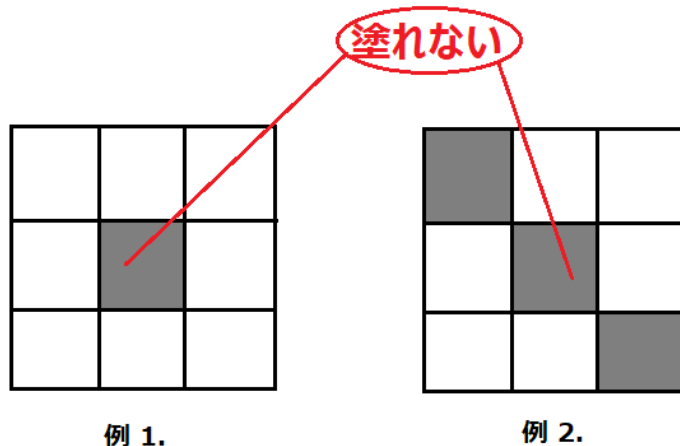
また、よく考えてみると、この方法では操作で選ぶのはすべて 1 つの整数に限られます。これは A, B, C の中で最大の整数なので、 $A + B + C + \max(A, B, C) \times (2^K - 1)$ が答えとなります。

ソースコード (C++) : <https://beta.atcoder.jp/contests/abc096/submissions/2456285>

ソースコード (Java) : <https://beta.atcoder.jp/contests/abc096/submissions/2457147>

C – Grid Repainting 2

以下の図のような「上下左右に隣り合うマスに黒いマスがない, 黒いマス」がある場合を考えましょう.



square1001 君は, あるマスを黒で塗る際に隣り合ったマスも一緒に黒で塗らなければならないので, 他の隣り合ったマスを塗らずにある特定のマスのみを塗るのは不可能です. ですので, このような場合における答えは必ず “No” となります.

逆に, それ以外の場合, どの黒いマスに対しても「上下左右に隣り合うマスのうち最低 1 つは黒」という条件を満たすこととなります. その場合, 全ての黒いマス A に対して, 「黒いマス A と, A と隣り合う黒いマスのうちの 1 つ B を両方塗る」という操作をすれば, 白にすべきマスを黒にすることなく, 黒にすべきマスをすべて黒にすることができます. よって, この場合答えは “Yes” となります.

「上下左右に隣り合うマスに黒いマスがない, 黒いマス」が存在するかどうか判定すればよいので, 個の問題は計算量 $O(HW)$ で解くことができます.

ソースコード (C++) : <https://beta.atcoder.jp/contests/abc096/submissions/2447972>

D – Five, Five Everywhere

出力すべき数列の条件はかなり厳しく、例えば「小さい順から、条件を満たすように選んでいく」という手法では $N = 9$ までしか答えを出すことはできません（ちなみに、 $a = [2, 3, 5, 7, 11, 13, 59, 1597, 34019]$ が作れる）。全探索のような手法でも難しそうなので、何か別の特別な方法を思いつかなければなりません。

想定解法

数列に含まれる数のうちすべてが「5 で割って 1 余る」数であれば、ここから 5 個の整数を選んだ時にこれらの合計は必ず 5 で割り切れます。また数列の値が素数、すなわち 2 以上の値であることから、5 つの整数の合計は必ず $2 \times 5 = 10$ 以上になります。5 で割り切れる正の整数の中で素数なものは 5 以外にないので、「5 で割り切れる 10 以上の整数」は、必ず合成数になります。

したがって、「5 で割って 1 余る、55555 以下の素数」を N 個列挙して、これを答えの数列にすればよいです。その場合、 $a = [11, 31, 41, 61, 71, 101, 131, 151, \dots, 1381]$ という数列を作ることができます。これは「すべての数は 55 555 以下」という制約を余裕で (!) 満たします。

また、「5 で割って 1 余る」だけではなく、余りの値を 1, 2, 3, 4 のいずれに設定しても条件を満たす数列を作ることができます。

ソースコード (C++) : <https://beta.atcoder.jp/contests/abc096/submissions/2444746>