Atcoder Beginner Contest 088 解説

問題作成者：E869120, square1001

平成 30 年 2 月 18 日

For International Readers: English editorial starts from page 5.

A: Infinite Coins

500 円硬貨が無限枚あるので，使う 1 円硬貨を最小化することを考えます．

N 円を超えないようにできるだけ多くの 500 円硬貨を使うと 1 円硬貨の枚数は最小化されます．そのとき，500 円硬貨は「N を 500 で割ったときの商」枚使われることになります．

この場合、使用する 1 円硬貨の枚数は N を 500 で割ったあまりになるので、これが持っている N 円硬貨の枚数 A 以下であれば “Yes”，そうでなければ “No” と出力します．

サンプルコード (C++)

- https://beta.atcoder.jp/contests/abc088/submissions/2105041

B: Card Game for Two

それぞれのプレイヤーは，自分のターンのときに，今選べるカードの中で最大の数が書かれているものを取るという戦略をとると，自分の得点を最大化できます．

a の値を大きい順に $a_1, a_2, \ldots, a_N$ となるように並べ替えるとき，$a_1 + a_3 + a_5 + a_7 + \cdots$ が Alice の得点となり，$a_2 + a_4 + a_6 + a_8 + \cdots$ が Bob の得点となります．

数列を小さい順・大きい順に並べ替える操作は「ソート (sorting)」といい，ほとんどのプログラミング言語において数列のソートをする関数はあります．存在しない場合でも，バブルソートなどの方法を使って数列をソートすることもできます．

サンプルコード (C++)：

- https://beta.atcoder.jp/contests/abc088/submissions/2105068

C: Takahashi's Information

高橋君の情報が正しい場合を考えます．$(a_1, a_2, a_3, b_1, b_2, b_3) = (p_1, p_2, p_3, q_1, q_2, q_3)$ の値の組み合わせが条件に沿っているならば，任意の数 x に対して，$(a_1, a_2, a_3, b_1, b_2, b_3) = (p_1 + x, p_2 + x, p_3 + x, q_1 - x, q_2 - x, q_3 - x)$ の値の組み合わせが正しいです．なぜなら，$p_i + q_j = (p_i + x) - (q_j + x)$ が成立するので，$a_i + b_j$ の値はどの (i, j) に対しても変化しないからです．

高橋君の情報が正しい場合，x を適切な値に設定することで，$a_1 = 0$ にすることができます．したがって，高橋君の情報が正しいかどうか判定するとき $a_1 = 0$ を仮定して考えてもよいです．a_1 の値が決まると，b_1, b_2, b_3 の値も決まります．具体的には，$b_i = c_{1,i} - a_1$ とすればよいです．b_1 の値が決まると，$a_i = c_{i,1} - b_1$ を利用して a_2, a_3 の値も決めることができます．

最終的に得られた $a_1, a_2, a_3, b_1, b_2, b_3$ の値に対して，$c_{i,j} = a_i + b_j$ が成立していれば高橋君の情報が正しく，そうでない場合高橋君の情報は間違っています．

D: Grid Repainting

まず，「けぬす君」が移動するルートがあらかじめ決まっている時，すぬけ君は最大で何マス変更できるかについて考えます．

最初に考えるべきは，各マスについてどのような操作ができるかです．「けぬす君」が通るマス・通らないマスそれぞれについて、何の色にすべきかという条件は，以下のようになります。

1. 「けぬす君」が通るマスは、絶対に白でなければならない．
2. 「けぬす君」が通らないマスは，白と黒のどちらでも良い．

まず，マスの色を黒から白に変更することはできないので，1．よりすぬけ君の通るルートに「最初の色」が黒であるマスが存在してはならず，すぬけ君の通るルートに含まれる「最初の色が白のマス」を黒に変えると 1．に反するので，これらのマスは変更することができません．

また，「けぬす君」が通らないマスにおける変更回数の合計の最大値は，2．より最終的な色は何でも良いため，けぬす君の通らない白いマスの色をすべて黒に変更することができるので，（けぬす君が通らない白いマスの個数）になります．

したがって，すぬけ君が変更できるマスの個数の最大値は，(けぬす君が通らない白いマスの個数) = **(マス目全体の白いマスの個数) − (けぬす君が通る白いマスの個数)** となります．

実際の問題では，けぬす君はマス (H,W) に白いマスだけを通ってたどり着くことができればどんなルートをたどっても良いです．しかし，「変更する回数」＝「色を変更できる白いマスの個数」を最大化するので，すなわち**「色を変更できない白いマスの個数」を最小化したい**です．

「色を変更できない白いマスの個数」は，(けぬす君が通る白いマスの個数) と等しいので、「けぬす君は $(1,1)$ から (H,W) に行くために最低何回の白いマスを通る必要があるか」という問題を解けば，この問題は解けます．言い換えると「けぬす君が白いマスだけを通って移動するとき，最小で何マスを使って $(1,1)$ から (H,W) に移動できるか」という問題になります．(このような経路が存在しない場合，ゲームクリアすることはできない)

この問題は辺の重みがすべて 1 であるグラフの最短経路問題に帰着することができるので，幅優先探索を用いれば，計算量 $O(HW)$ で解くことができます．

Atcoder Beginner Contest 088 Editorial

Problem Setters : E869120, square1001

平成 30 年 2 月 18 日

A: Infinite Coins

There are infinite number of 500-yen coins. So, to minimize the number of 1-yen coins, he has to maximize the number of 500-yen coins. In this case the number of 500-yen coin is “the quotient when N is divided by 500”.

In this case, the number of 1-yen coins is “the remainder when N is divided by 500”, so you should check if the remainder when N is divided by 500 is greater than A or not.

Sample Code (C++)

- https://beta.atcoder.jp/contests/abc088/submissions/2105041

B: Card Game for Two

If the player takes a strategy which chooses a card with maximum number among all selectable cards in each turn, the final score will be maximized.

Now, let's rearrange the sequence a, that $a_1, a_2, a_3, \ldots, a_N$ will be in decreasing order. Alice's score will be $a_1 + a_3 + a_5 + a_7 + \cdots$, and Bob's score will be $a_2 + a_4 + a_6 + a_8 + \cdots$. Therefore, the difference of Alice's score and Bob's score will be $a_1 - a_2 + a_3 - a_4 + a_5 - a_6 + a_7 - a_8 + \cdots$.

Rearranging sequence in increasing order or decreasing order is called "sorting". Sorting is the basic algorithm in computer science, and most programming language has sorting function. If you don't use programming language that doesn't have sorting function, you can implement sorting algorithm like Bubble Sort.

Sample Code (C++) :

- https://beta.atcoder.jp/contests/abc088/submissions/2105068

C: Takahashi's Information

If Takahashi's information is correct, there exists a combination of numbers $(a_1, a_2, a_3, b_1, b_2, b_3) = (p_1, p_2, p_3, q_1, q_2, q_3)$ that satisfies $a_i + b_j = c_{i,j}$ for all (i, j). And in this case, for any number x, $(a_1, a_2, a_3, b_1, b_2, b_3) = (p_1 + x, p_2 + x, p_3 + x, q_1 - x, q_2 - x, q_3 - x)$ also satisfies $a_i + b_j = c_{i,j}$, because $p_i + q_j = (p_i + x) + (q_j - x)$ holds.

So, if Takahashi's information is correct, you can set $a_1 = 0$ if you choose number x properly. Therefore, if you only want to judge whether Takahashi's information is correct, deciding "$a_1 = 0$" is okay. Then, you can get the values of b_1, b_2, b_3, because $b_i = c_{1,i} - a_1$ therefore you can set $b_i = c_{1,i}$. And then, if you have the value of b_1, you can get the values of a_2, a_3 because $a_i = c_{i,1} - b_i$ holds.

Now you got the values of $a_1, a_2, a_3, b_1, b_2, b_3$. Finally, Takahashi's information is correct if and only if $a_i + b_j = c_{i,j}$ holds for all (i, j) so you can check it easily.

D: Grid Repainting

At first, let's think about the maximum number of cells that Snuke changes, when the route Kenus moves is already decided.

In this case, to achieve the maximum number of cells that Snuke changes, Snuke should change cells as follows:

a. If the cell is belonged to the Kenus's route, the cell must be white. Thus, it is not valid route if at least one of cell is black at first because the color of black cell cannot be changed. And, if it is white, it should remain white so Snuke cannot change the color of cells in this route.
b. If the cell is not belonged to the Kenus's route, any color is OK. So, if the cell is white, Snuke should change the color of the cell (from white to black) to achieve maximum number of cells that he changes.

Therefore, the maximum number of cells that Snuke changes is (The number of white cells that are not belonged to the Kenus's route), and this is equal to **(The number of white cells in the grid) – (The number of white cells that are belonged to the Kenus's route).**

Finally, the first easier problem was solved. But unfortunately, the real problem is: Kenus's route is not decided, and Snuke can decide it.

Snuke want to minimize the number of white cells that are belonged to the Kenus's route. In addition, the number of white cell that are belonged to the Kenus's route is equal to the number of cells that Kenus passes, because all cell that Kenus passes should be white.

So, now the problem is simple. The minimum number of white cells Kenus passes is equal to the result of the problem: "You are given a grid and each cell is black or white. You move from $(1,1)$ to (H,W) but you can't pass black cells. If you can move one cell left, right, up or down, what is the minimum number of cell you pass?".

This is a typical problem of "shortest path problem". You can use Breadth-First Search (BFS) and solvable in $O(HW)$ time complexity.