ABC 121 解説

writer: drafear

2019 年 3 月 9 日

A: Cells

”残る白色のマス目の数は行や列の選び方によらない” ことを信じると、例えば後ろから h 行 w 列を選んで黒色で塗りつぶした場合の白マス数を数えれば良いです。このときの白マス数は $(H-h)\times(W-w)$ です。

他の考え方もあります。初めに h 行選んで黒く塗りつぶし、次に w 列選んで黒く塗りつぶすことを考えます。すると、初めに h 行選んで黒く塗りつぶされる白マス数は $h\times W$ になります。その後に w 列選んで黒く塗りつぶされる白マス数は、1 列選ぶたびに $H-h$ 個の白マスが塗りつぶされるので $(H-h)\times w$ になります。初め、白マスは $H\times W$ 個あったので、答えは $HW-hW-(H-h)w$ になります。

”残る白色のマス目の数は行や列の選び方によらない” ことの直感的な証明を述べます。選ばれなかった行、列の数はそれぞれ $H-h$ 行、$W-w$ 列です。選ばれなかった各行、各列に対して 1 つの白マスが対応するので、答えは $(H-h)\times(W-w)$ になります。

C++ での実装例を以下に挙げます。

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

using namespace std;

int main() {
  int H, W; cin >> H >> W;
  int h, w; cin >> h >> w;
  int ans = (H - h) * (W - w);
  cout << ans << endl;
}
```

B: Can you solve this?

各 i について、$sum(i) = A_{i1}B_1 + A_{i2}B_2 + ... + A_{iM}B_M$ を計算し、$sum(i) + C > 0$ であるような i の個数が答えになります。これを実装できれば正答することができますが、実装に慣れていない場合特に入力を受け取る部分の実装が難しいと思います。以下の実装例では、

1. 入力を受け取り、変数に格納する
2. 計算し、答えを求める
3. 答えを出力する

の 3 つのパートに分けて実装しています。他の問題でも、入力・計算・出力 に分けて実装するのが分かりやすいのでおすすめです。今回の場合、変数の型としては、A は整数型の 2 次元配列、B は整数型の 1 次元配列、N, M, C は整数型とするのが良いでしょう。

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

using namespace std;

int main() {
  // input
  int N, M, C; cin >> N >> M >> C;
  vector<int> B(M);
  for (int i = 0; i < M; ++i) {
    cin >> B[i];
  }
  vector<vector<int>> A(N, vector<int>(M));
  for (int i = 0; i < N; ++i) {
    for (int j = 0; j < M; ++j) {
      cin >> A[i][j];
    }
  }
  // compute
  int ans = 0;
  for (int i = 0; i < N; ++i) {
    // check whether i-th source code solves this problem
    int sum = 0;
    for (int j = 0; j < M; ++j) {
      sum += A[i][j] * B[j];
    }
    if (sum + C > 0) {
```

```
      ++ans;
    }
  }
  // output
  cout << ans << endl;
}
```

C: Energy Drink Collector

安い店から順に M 本のエナドリを買うのが最善です。したがって、A_i の小さい順にソートして、M 本に達するまで前から順に買えるだけ買うことをシミュレーションすれば答えを求めることができます。ソートの方法はいくつかありますが、例えば C++ では vector<pair<int,int>>に組 (A_i, B_i) を全て入れて sort 関数を呼ぶことでソートすることができます。計算量は、ソート部分がボトルネックで $O(NlogN)$ です。

D: XOR World

x と y の排他的論理和を $x \oplus y$ と書くことにします。排他的論理和の計算順序は入れ替えて良く、また、どの非負整数 n についても $n \oplus n = 0$ なので、$A > 0$ のとき

$$F(A, B) = F(0, A-1) \oplus F(0, B)$$

が成り立ちます。そこで、$f(n) = F(0, n)$ を求めることを考えます。

これは、以下の重要な性質に気づくことができれば、簡単に求めることができます。気づかなかった場合でも、答えを 2 進数で表したときの上位桁から順に決めていく $O(log(B))$ の解法もあります。

重要な性質

任意の偶数 n について $n \oplus (n+1) = 1$

この性質は、偶数 n について n と $n+1$ を 2 進数表記したときに最下位ビットのみが異なることから分かります。例えば、$100 = (1100100)_2, 101 = (1100101)_2$ です。この性質を使えば、例えば

$$\begin{aligned} & 0 \oplus 1 \oplus 2 \oplus 3 \oplus 4 \oplus 5 \oplus 6 \\ = & (0 \oplus 1) \oplus (2 \oplus 3) \oplus (4 \oplus 5) \oplus 6 \\ = & 1 \oplus 1 \oplus 1 \oplus 6 \end{aligned}$$

のように計算できます。$1 \oplus 1 \oplus ... \oplus 1$ は 1 が偶数個なら 0、奇数個なら 1 になります。このように計算することで、答えを $O(1)$ で求めることができます。