CodeFlyer 予選 解説

wo01

2018 年 6 月 2 日

A: 本選参加者数

この問題は、A 以下の最大の B の倍数を求める問題です。
以下に C++ による実装例を示します。

```
#include<cstdio>

using namespace std;

int main(){
    int A, B;
    scanf("%d%d", &A, &B);
    printf("%d\n", A / B * B);
    return 0;
}
```

B: 洋菓子店

残っているショートケーキ、チーズケーキの個数を保持しながらシミュレーションすればよいです。
以下に C++ による実装例を示します。

```
#include<cstdio>
#include<cstring>

using namespace std;

const int MAX_N = 100000;

int A, B;
int N;
char ch[MAX_N + 1];
```

```
int main(){
    scanf("%d%d", &A, &B);
    scanf("%d", &N);
    scanf("%s", ch);
    for(int i = 0; i < N; ++i){
        if(ch[i] == 'S'){
            if(A > 0) A--;
        }else if(ch[i] == 'C'){
            if(B > 0) B--;
        }else{
            if(A >= B && A > 0) A--;
            else if(B > 0) B--;
        }
    }
    printf("%d\n%d\n", A, B);
    return 0;
}
```

C: 徒歩圏内

以下の値 A, B が求められれば、前者から後者を引くことで答えが求められます。

1. $i < j < k$ かつ $X_k - X_i \leq D$ を満たす (i, j, k) の個数 A
2. $i < j < k$ かつ $X_j - X_i \leq D$, $X_k - X_i \leq D$ を満たす (i, j, k) の個数 B

なお、(i, j, k) が 1 の条件を満たすとき、$X_j - X_i \leq D$, $X_k - X_j \leq D$ も満たされていることに注意してください。

これらの値の求め方を考えましょう。

まず準備として、各 i について以下のように $\mathit{left}(i)$, $\mathit{right}(i)$ を定義します。

$$\mathit{left}(i) = (j < i \text{ かつ } X_i - X_j \leq D \text{ を満たす } j \text{ の個数})$$
$$\mathit{right}(i) = (j > i \text{ かつ } X_j - X_i \leq D \text{ を満たす } j \text{ の個数})$$

各 i について二分探索を行うことで、$\mathit{left}(1), \mathit{left}(2), \ldots, \mathit{left}(N)$ および $\mathit{right}(1), \mathit{right}(2), \ldots, \mathit{right}(N)$ のすべての値を $O(N \log N)$ 時間で求めることができます（全体で $O(N)$ 時間で求める方法もあります)。

これらの値を用いると、求めたかった値 A, B は以下のようにして求められます。

- 値 A の求め方
  条件「$i < j < k$ かつ $X_k - X_i \leq D$」に加え $i = l$ を満たす (i, j, k) の個数は $\binom{\mathit{right}(l)}{2}$ です[*1]。これを各 l について求め、足し合わせることで A の値が求まります。

[*1] $n < k$ のとき $\binom{n}{k} = 0$ とします

- 値 B の求め方
  条件「$i < j < k$ かつ $X_j - X_i \leq D,\ X_k - X_i \leq D$」に加え $j = m$ を満たす (i, j, k) の個数は $\mathit{left}(m) \times \mathit{right}(m)$ です。これを各 m について求め、足し合わせることで B の値が求まります。

D: ハンコ

まず、ハンコを「マスごとに分解」することが重要です。ハンコのマス (i, j) に当たる部分が黒色のとき、問題文の操作を行うと紙のマス (i, j) を左上、マス $(H - i + 1, W - j + 1)$ を右下とする長方形領域に含まれるすべてのマスが黒色になります。このことから、もとの問題は以下のように言い換えられます。

> $H \times W$ のマス目がある。文字列 A_i の j 文字目が `'#'` のとき、マス (i, j) を左上、マス $(H - i + 1, W - j + 1)$ を右下とする長方形領域を黒く塗る。最終的にいくつのマスが黒く塗られるか。

この問題の解き方を考えましょう。

黒く塗られる長方形領域の上端または下端として現れる行は $1, 2, \ldots, N$ および $H - N + 1, H - N + 2, \ldots, H$ 行目だけです。よって、$N + 1, N + 2, \ldots, H - N$ 行目を（存在すれば） 1 行に圧縮することができます。列についても同様に圧縮することで、最大 $(2N + 1) \times (2M + 1)$ のマス目が得られます。

文字列 A_i の j 文字目が `'#'` となるような各 i, j について、圧縮後のマス目におけるマス (i, j) を左上、マス $(H - i + 1, W - j + 1)$ を右下とする長方形領域にあたる部分を黒く塗る操作を行うことを考えます。これは、累積和のテクニックを用いることで全体で $O(NM)$ で実現できるので、高速に答えが求まります。

補足: 累積和のテクニック

> $N \times M$ の 2 次元配列 A があり、はじめすべての要素の値が 0 であるとします。$i_1, i_1 + 1, \ldots, i_2$ 行 $j_1, j_1 + 1, \ldots, j_2$ 列の長方形領域に含まれる各要素に 1 を足すことを考えます。この操作は以下のアルゴリズムにより実現できます。
>
> 1. `A[i1][j1]`, `A[i2 + 1][j2 + 1]` に 1 を、`A[i1][j2 + 1]`, `A[i2 + 1][j1]` に -1 を足す。
> 2. 各 i について、$j = 0, 1, \ldots$ の順に、`A[i][j + 1] += A[i][j]` という処理をする。
> 3. 各 j について、$i = 0, 1, \ldots$ の順に、`A[i + 1][j] += A[i][j]` という処理をする。
>
> このアルゴリズムは長方形領域が Q 個ある場合にも拡張でき、全体の計算量は $O(Q + NM)$ となります。

E: 祝日

まず、$N = 0$ の場合について考えます。

値 x_{ij} を、一年の最初の日の曜日が i である場合の B_j 回目の曜日 C_j の日が一年の x_{ij} 日目となるように定めます。各 i に対して、$x_{i1}, x_{i2}, \ldots, x_{iM}$ をソートしたときの隣り合う要素の差に関する情報を知りたいことになります。

x_{ij} と $x_{i+1,j}$ の関係は以下のようになります。

$$x_{i+1,j} = \begin{cases} x_{ij} + W - 1 & (\text{if } C_j = i) \\ x_{ij} - 1 & (\text{otherwise}) \end{cases}$$

多くの i, j について $x_{i+1,j} = x_{ij} - 1$ が成り立つことに注目して、値 y_{ij} を以下のように定めます。

$$y_{ij} = x_{ij} + i$$

このとき、y_{ij} と $y_{i+1,j}$ の関係は以下のようになります。

$$y_{i+1,j} = \begin{cases} y_{ij} + W & (\text{if } C_j = i) \\ y_{ij} & (\text{otherwise}) \end{cases}$$

各 i について、$x_{i1}, x_{i2}, \ldots, x_{iW}$ をソートしたときの隣り合う要素の差の多重集合と $y_{i1}, y_{i2}, \ldots, y_{iW}$ をソートしたときの隣り合う要素の差の多重集合は等しくなります。

以上から、$N = 0$ が成り立つ場合、この問題は次のようにして解くことができます。

1. $Y = \{y_{11}, y_{12}, \ldots, y_{1W}\}$ を求め、$d = 1$ のときの答え ans_1 を求める。
2. $d = 1, 2, \ldots, W - 1$ について以下を行う
   - $B_j = d$ となる各 j について、Y から y_{dj} を削除し、$y_{d+1,j}$ を追加する。その際、Y における y_{dj} と $y_{d+1,j}$ の前後の値と ans_d の値から ans_{d+1} の値を計算する。

$N \leq 50$ の場合への拡張は容易なので省略します。日付による祝日と曜日による祝日が重なる場合の処理に注意してください。