AISing Programming Contest 2019 解説

wo01

2019 年 1 月 12 日

A 問題

答えの値は $(N - H + 1) \times (N - W + 1)$ となります。

以下に C++ による実装例を示します。

```
#include<cstdio>

using namespace std;

int N;
int H, W;

int main(){
	scanf("%d", &N);
	scanf("%d%d", &H, &W);
	printf("%d\n", (N - H + 1) * (N - W + 1));
	return 0;
}
```

https://atcoder.jp/contests/aising2019/submissions/3972260

B 問題

答えの値は次の 3 つの値のうち最も小さいものとなります。

- $P_i \leq A$ を満たす i の個数
- $A + 1 \leq P_i \leq B$ を満たす i の個数
- $P_i \geq B + 1$ を満たす i の個数

これらの値はループ（または他の言語機能）を用いて求められます。

以下に C++ による実装例を示します。

```
#include<cstdio>
#include<algorithm>

using namespace std;

const int MAX_N = 100;

int N;
int P[MAX_N];

int A, B;

int main(){
    scanf("%d", &N);
    scanf("%d%d", &A, &B);
    for(int i = 0; i < N; ++i) scanf("%d", P + i);
    int e = 0, m = 0, h = 0;
    for(int i = 0; i < N; ++i){
        if(P[i] <= A) e++;
        else if(P[i] <= B) m++;
        else h++;
    }
    int ans = min(min(e, m), h);
    printf("%d\n", ans);
    return 0;
}
```

https://atcoder.jp/contests/aising2019/submissions/3972264

C 問題

各マスを頂点とし、隣り合うマスの組であって色が異なるものの間に辺を張ったグラフを考えます。

このグラフに連結成分が k 個あり、このうち i 番目のものは b_i 個の黒マスと w_i 個の白マスからなっていたとします。このとき、答えは $\sum_{1 \leq i \leq k} b_i w_i$ となります。

あとは $(b_1, w_1), (b_2, w_2), \ldots, (b_k, w_k)$ の値を求めれば良いことになりますが、これはグラフ上の (BFS または DFS での) 探索、または Union-Find 木により求められます。

実装例: `https://atcoder.jp/contests/aising2019/submissions/3972253`

D 問題

N は偶数であるとします（奇数の場合もほぼ同様です）。また、整数 A_i が書かれたカードをカード i と呼ぶことにします。

まず、各 i について以下の値を前計算しておきます。

$$sum_i = A_1 + A_2 + \cdots + A_i$$
$$evenSum_i = A_2 + A_4 + \cdots + A_{i'}$$

ただし i' は i 以下の最大の偶数です。

X を一つ固定し、$x = X$ のときについて考えます。

カードの二人への分割はある i, j を用いて以下のように書けます。

- カード $1, 2, \ldots, i$ については、高橋くんが偶数番号のカードを、青木くんが奇数番号のカードを取る。
- カード $i+1, i+2, \ldots, j$ はすべて青木くんが取る。
- カード $j+1, j+2, \ldots, N$ はすべて高橋くんが取る。

このような i, j を高速に求められれば、前計算しておいた値を用いて答えが求められます[*1]。

このような i, j は二分探索を用いて求められます。詳細な方針はいろいろありそうですが、ここでは以下の実装例における方針を少し詳しく説明します。

添字 $1, 2, \ldots, N$ を A_i の値が x に近い方から順に並べると $s_1, s_2, \ldots, s_N$ となるとします。以下を満たす最大の t の値 T を求められれば、i, j の値は求められます。

$$\{s_1, s_2, \ldots, s_{\lfloor t/2 \rfloor}\} \cap \{N, N-1, \ldots, N - \left\lceil \frac{t}{2} \right\rceil + 1\} = \emptyset$$

各 t に対して、ある l_t, r_t が存在して、

$$\{s_1, s_2, \ldots, s_{\lfloor t/2 \rfloor}\} = \{l_t, l_t + 1, \ldots, r_t\}$$

となります。この l_t の値と $N - t + 1$ の値の大小関係は t と T の大小関係だけに依存し、両者が一致するとき $t = T$ となります。

したがって、l_t と $N - t + 1$ の値の大小関係がわかれば良いことになりますが、これは A_{N-t} と $A_{N-\lceil t/2 \rceil}$ のどちらが x に近いかを調べることでわかります。

実装例: `https://atcoder.jp/contests/aising2019/submissions/3972256`

[*1] $A_{j+1} + A_{j+2} + \cdots + A_N = sum_N - sum_j$ のような式変形を用います

E 問題

入力の木を頂点 1 を根とする根付き木とみなします。

この問題は以下のような動的計画法 (DP) によって解くことができます*2 。

- $dp_{1,v,k} =$ (頂点 v 以下の部分木を問題文の条件の通り k 個の成分にわけ、頂点 v を含む成分の頂点の A の値はすべて正になるようにするときの頂点 v を含む成分の A の値の和の最小値)
- $dp_{2,v,k} =$ 頂点 v 以下の部分木を、k 個の成分にわけ、頂点 v を含む成分以外は問題文の条件を満たすようにするときの頂点 v を含む成分の A の値の和の最小値)

ただし分割が不可能なときは対応する DP 配列の値は十分大きい値 inf とします。

この DP は各頂点 v において以下のような補助的な DP を行うことで求められます。

- $dp'_{1,i,j} =$ (頂点 v の i 番目の子までからなる部分木を $dp_{1,v,k}$ と同様の条件のもとで j 個の成分に分けたときの頂点 v を含む成分の A の値の和の最小値)
- $dp'_{2,i,j}$ も同様

この DP の計算量を考えましょう。状態数が $O(N)$ であり、各状態の更新に最悪 $O(N)$ 時間かかるので全部で $O(N^3)$ 時間かかるように見えますが、よく観察すると全部で $O(N^2)$ 時間で動作することがわかります*3。

答えは $dp_{1,1,k} \neq inf$ または $dp_{2,1,k} < 0$ を満たす最小の k の値から 1 を引いたものです。

実装例: `https://atcoder.jp/contests/aising2019/submissions/3972259`

---

*2 以下のような DP を思いついたかもしれません。$dp_{i,v,x} =$ (頂点 v 以下の部分木を解説本文と同様にいくつかの成分に分け、頂点 v を含む成分の頂点の A の値の和が x になるようにするときの成分の個数の最小値)、ただし $i = 1, 2$

解説本文中の DP はこの DP の key と value を入れ替えたものとなっています。

このように key と value を入れ替えるというテクニックは汎用的です

*3 「木　二乗 DP」などのキーワードで検索してみてください