ARC093 / ABC092 解説

wo01

2018 年 3 月 25 日

A 問題

答えは $\min(A, B) + \min(C, D)$ となります。
以下に C++ による実装例を示します。

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

using namespace std;

int main(){
    int A, B, C, D;
    scanf("%d%d%d%d", &A, &B, &C, &D);
    printf("%d\n", min(A, B) + min(C, D));
    return 0;
}
```

B 問題

i 人目の参加者は合宿中にチョコレートを $1 + \lfloor (D-1)/A_i \rfloor$ 個食べます。ここで、$\lfloor x \rfloor$ は x を超えない最大の整数です。

よって、答えは

$$X + \sum_{1 \le i \le N} (1 + \left\lfloor \frac{D-1}{A_i} \right\rfloor)$$

となります。

以下に C++ による実装例を示します。C++ を含む多くの言語で、正の整数同士の除算の際はあまりが切り捨てられます。

```
#include<cstdio>

using namespace std;

const int MAX_N = 100;

int N;
int A[MAX_N];
int D;
int X;

int main(){
    scanf("%d", &N);
    scanf("%d%d", &D, &X);
    for(int i = 0; i < N; ++i){
        scanf("%d", A + i);
    }
    int ans = X;
    for(int i = 0; i < N; ++i){
        int tmp = (D - 1) / A[i] + 1;
        ans += tmp;
    }
    printf("%d\n", ans);
    return 0;
}
```

C 問題

入力で与えられる $A_1, A_2, \ldots, A_N$ に加えて、$A_0 = A_{N+1} = 0$ としておきます。求めたいものは、各 i $(1 \leq i \leq N)$ に対する

$$\begin{aligned} S_i = &|A_0 - A_1| + |A_1 - A_2| + \cdots + |A_{i-2} - A_{i-1}| + |A_{i-1} - A_{i+1}| + \\ &|A_{i+1} - A_{i+2}| + \cdots + |A_N - A_{N+1}| \end{aligned}$$

の値です。

ここで、

$$S = \sum_{0 \leq j \leq N} |A_j - A_{j+1}|$$

とします。この値は $O(N)$ 時間で求められます。

このとき、

$$S_i = S + |A_{i-1} - A_{i+1}| - (|A_{i-1} - A_i| + |A_i - A_{i+1}|)$$

となります。よって、S_i の値は各 i に対して $O(1)$ 時間で求められます。

以上から、全体で $O(N)$ 時間で解くことができます。

D 問題

条件を満たすグリッドは以下のようにして構成することができます。

- $K = 50$ として、$2K \times 2K$ のグリッドを用意する。
- 上から K 行以内にあるマスを全て黒く塗り、残りのマスをすべて白く塗る。
- 上から $K-1$ 行以内のマスのうち $A-1$ 個のマスを上下左右斜めに隣り合わないように選び、それらの色を白に変える。
- 下から $K-1$ 行以内のマスのうち $B-1$ 個のマスを上下左右斜めに隣り合わないように選び、それらの色を黒に変える。

E 問題

与えられたグラフを G とし、 G の最小全域木 T を一つ選びます。また、その重みを S とします。

T の 2 頂点 u, v に対して、$pathMax(u,v)$ を T の uv パス上の辺の重みの最大値と定義します。さらに、G に含まれるが T に含まれない辺 $e = \{u, v\}$ が重み c を持つとき、$diff(e) = c - pathMax(u,v)$ と定義します。T が最小全域木であることから、すべての e に対して $diff(e) \geq 0$ となります。

さて、G の辺を白色または黒色に塗ったとき、白く塗られた辺と黒く塗られた辺をともに含む全域木のうち最小の重みのものの重み S' は以下のようになります。

- T に白色の辺と黒色の辺がともに含まれる場合、$S' = S$
- そうでない場合、T の辺と異なる色で塗られた辺のうち最小の $diff$ の値を持つものを e として、$S' = S + diff(e)$

このことから、求めたい塗り方の個数は以下のようにして求められます。

まず、$D = X - S$ とします。さらに、$diff(e) < D$, $diff(e) = D$, $diff(e) > D$ を満たす e の個数をそれぞれ $lower$, $equal$, $upper$ とします。このとき、

- $D < 0$ のとき、塗り方の個数は 0
- $D = 0$ のとき、
  - T の辺に 2 種類の色を両方用いる場合、塗り方の個数は $(2^{N-1} - 2)2^{M-N+1}$
  - そうでない場合、塗り方の個数は $2 \times (2^{equal} - 1) \times 2^{upper}$
- $D > 0$ のとき、塗り方の個数は $2 \times (2^{equal} - 1) \times 2^{upper}$

以上で答えが求まりました。

F 問題

条件を満たす p のうち $p_1 = 1$ であるものの個数がわかれば十分です。以下、$p_1 = 1$ を仮定します。

選手 1 が優勝または敗退するまでに対戦する可能性がある選手の番号は

$$p_2, \min\{p_3, p_4\}, \ldots, \min\{p_{2^{N-1}+1}, \ldots, p_{2^N}\}$$

です。上の N 個の値がどれも集合 $\{A_1, A_2, \ldots, A_M\}$ に含まれないとき、選手 1 が優勝します。

選手 $2, 3, \ldots, 2^N$ を番号順に**場所** $p_2, p_3, \ldots, p_{2^N}$ に割り当てていくことを考えます。また、これらの場所は N 個のグループ $G_1 = \{p_2\}$, $G_2 = \{p_3, p_4\}$, ..., $G_N = \{p_{2^{N-1}+1}, \ldots, p_{2^N}\}$ に分かれているとします。グループ G_i が**開始済みグループ**であるとは、グループ G_i に属する選手が 1 人以上決まっていることを言うことにします。また、$\{1, 2, \ldots, N\}$ の部分集合 U に対して、

$$\mathrm{sz}(U) = \sum_{i \in U} 2^{i-1}$$

と定めます。

次のような DP を考えます。

$$\begin{aligned}
dp_{1,i,U} &= \text{選手 } 2, 3, \ldots, A_i \text{ の場所を決める方法であって、} \\
&\quad\ \text{開始済みグループの集合 } T \text{ が } T = U \text{ を満たすものの個数} \\
dp_{2,i,U} &= \text{選手 } 2, 3, \ldots, A_i \text{ の場所を決める方法であって、} \\
&\quad\ \text{開始済みグループの集合 } T \text{ が } T \subseteq U \text{ を満たすものの個数} \\
dp_{3,i,U} &= \text{選手 } 2, 3, \ldots, A_i - 1 \text{ の場所を決める方法であって、} \\
&\quad\ \text{開始済みグループの集合 } T \text{ が } T \subseteq U \text{ を満たすものの個数} \\
dp_{4,i,U} &= \text{選手 } 2, 3, \ldots, A_i - 1 \text{ の場所を決める方法であって、} \\
&\quad\ \text{開始済みグループの集合 } T \text{ が } T = U \text{ を満たすものの個数}
\end{aligned}$$

この DP テーブルは以下のように計算することができます。

- 各 U に対して $dp_{1,i,U}$ が求まっているとき、各 U に対する $dp_{2,i,U}$ は高速ゼータ変換 (の変形) により求められます。
- 各 U に対して $dp_{2,i,U}$ が求まっているとき、各 U に対する $dp_{3,i+1,U}$ は、$t = \max(0, \mathrm{sz}(U) - A_i)$, $d = A_{i+1} - 1 - A_i$ として、

  $$dp_{3,i+1,U} = t \times (t-1) \times \cdots \times (t-d+1) \times dp_{1,i,U}$$

  として求められます。
- 各 U に対して $dp_{3,i,U}$ が求まっているとき、各 U に対する $dp_{4,i,U}$ は高速ゼータ変換 (の変形) により求められます。
- 各 U に対して $dp_{4,i,U}$ が求まっているとき、各 U に対する $dp_{1,i,U}$ は

  $$dp_{1,i+1,U} = (\mathrm{sz}(U) - (A_i - 1)) \times dp_{4,i,U}$$

  として求められます。

この DP テーブルが求められれば、あとは適切に足し合わせて適切な係数をかけるだけで答えが求められます。

以上で答えが求まりました。

ARC093 / ABC092 Editorial

wo01

March 25th, 2018

Problem A

The answer is $\min(A, B) + \min(C, D)$.

C++ example:

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

using namespace std;

int main(){
    int A, B, C, D;
    scanf("%d%d%d%d", &A, &B, &C, &D);
    printf("%d\n", min(A, B) + min(C, D));
    return 0;
}
```

Problem B

The i-th participant eats $1 + \lfloor (D-1)/A_i \rfloor$ chocolates. Here $\lfloor x \rfloor$ denotes the largest integer that doesn't exceed x.

Therefore, the answer is

$$X + \sum_{1 \le i \le N} (1 + \left\lfloor \frac{D-1}{A_i} \right\rfloor)$$

C++ example:

```
#include<cstdio>

using namespace std;

const int MAX_N = 100;

int N;
int A[MAX_N];
int D;
int X;

int main(){
    scanf("%d", &N);
    scanf("%d%d", &D, &X);
    for(int i = 0; i < N; ++i){
        scanf("%d", A + i);
    }
    int ans = X;
    for(int i = 0; i < N; ++i){
        int tmp = (D - 1) / A[i] + 1;
        ans += tmp;
    }
    printf("%d\n", ans);
    return 0;
}
```

Problem C

Let $A_0 = A_{N+1} = 0$. For each i ($1 \le i \le N$), we want to compute the following value:

$$S_i = |A_0 - A_1| + |A_1 - A_2| + \cdots + |A_{i-2} - A_{i-1}| + |A_{i-1} - A_{i+1}| + |A_{i+1} - A_{i+2}| + \cdots + |A_N - A_{N+1}|$$

Let

$$S = \sum_{0 \le j \le N} |A_j - A_{j+1}|$$

and we pre-compute this value in $O(N)$ time.

Then, since

$$S_i = S + |A_{i-1} - A_{i+1}| - (|A_{i-1} - A_i| + |A_i - A_{i+1}|)$$

we can compute the value of S_i in $O(1)$ for each i.

This solution works in $O(N)$ time in total.

Problem D

We can construct the grid as follows:

- Let $K = 50$ and prepare a grid of dimensions $2K \times 2K$.
- Paint all cells in the topmost K rows black. Paint the other cells white.
- From the topmost $K - 1$ rows, choose $A - 1$ cells such that no two chosen cells are 8-adjacent, and paint them white.
- From the bottommost $K - 1$ rows, choose $B - 1$ cells such that no two chosen cells are 8-adjacent, and paint them black.

Problem E

Let G be the given graph, and let T be (one of) its MSTs. Let S be its weight.

For two vertices u, v in T, define $pathMax(u, v)$ as the weight of the heaviest edge on the path uv along T. Also, if an edge $e = \{u, v\}$ is not contained in T, define $diff(e) = c - pathMax(u, v)$, where c is the weight of edge e. Note that for edges e in T, the value of $diff(e)$ is undefined (make sure not to regard them as zeroes). Since T is an MST, for all edge e outside T, $diff(e) \geq 0$ holds.

Suppose that we paint the edges of G black and white. The weight of the MST that contains both black and white edges can be computed as follows (let's denote it S'):

- If T contains both black and white edges, $S' = S$.
- If T contains only black edges, $S' = S + diff(e)$. Here, e is a white edge that minimizes the value of $diff(e)$.
- (Similarly, we can handle the case where all edges in T are white).

Therefore, the answer can be computed as follows.

First, let $D = X - S$. Let *lower*, *equal*, *upper* be the number of edges e such that $diff(e) < D$, $diff(e) = D$, and $diff(e) > D$, respectively.

Then,

- If $D < 0$, the answer is 0.
- If $D = 0$,
  - In case we use both colors at least once for the edges in T, there are $(2^{N-1} - 2)2^{M-N+1}$ ways to paint edges: $(2^{N-1} - 2)$ ways to color T by using both colors, and 2^{M-N+1} ways to arbitrarily color other edges.
  - Otherwise, there are $2 \times (2^{equal} - 1) \times 2^{upper}$ ways to paint edges: two ways to color edges in T, $(2^{equal} - 1)$ ways to color edges in "equal" (the color of at least one edge must be different from T's color), and 2^{upper} ways to arbitrarily color "upper" edges.
- If $D > 0$, the answer is $2 \times (2^{equal} - 1) \times 2^{upper}$: two ways to color edges in T and "lower", $(2^{equal} - 1)$ ways to color edges in "equal" (the color of at least one edge must be different from T's color), and 2^{upper} ways to arbitrarily color "upper" edges.

Problem F

Assume that $p_1 = 1$ (and later multiply the answer by 2^N).

Player 1 can win the tournament if he can beat all of the following players:

$$p_2, \min\{p_3, p_4\}, \dots, \min\{p_{2^{N-1}+1}, \dots, p_{2^N}\}$$

Thus, we want to compute the number of permutations such that none of the N numbers above are contained in the set $\{A_1, A_2, \dots, A_M\}$.

Consider N groups of positions: $G_1 = \{p_2\}$, $G_2 = \{p_3, p_4\}$, $\dots$, $G_N = \{p_{2^{N-1}+1}, \dots, p_{2^N}\}$. We want to count the number of permutations such that for each group g in $\{G_1, \dots, G_N\}$, the minimum element in g is *not* in $\{A_1, A_2, \dots, A_M\}$. Let's use inclusion-exclusion principle. For each subset S of $\{G_1, \dots, G_N\}$, we want to compute $f(S)$, the number of permutations such that

- For each group g in S, the minimum element of g is in $\{A_1, A_2, \dots, A_M\}$.

To compute these values, do the following DP. Assume that $A_1 > A_2 > \cdots > A_M$. For each $i = 1, \cdots, M$ in this order, do one of the following:

- For some integer k, choose $2^k - 1$ players greater than A_i, and assign them to the group G_k, together with the player A_i. Now the minimum element in G_k is in $\{A_1, A_2, \dots, A_M\}$.
- Do nothing.

For each integer $0 \le i \le M$ and a set $S \subseteq \{G_1, \dots, G_N\}$, define $dp_{i,S}$ as the number of ways such that when we process players $A_1, \cdots, A_i$ as described above, the groups in S are filled, and other groups are empty. Then, we can compute $f(S)$ as $dp_{M,S}$ times some coefficients, and by inclusion-exclusion principle, the answer is $\sum f(S)(-1)^{|S|}$.