Code Festival 予選 A 解説

wo01

2018 年 9 月 22 日

A: 配点

配点の合計は $A+B+C$ 以上 $A+B+C+3$ 以下の整数となります。また、これら 4 個の整数はすべて配点の合計として実現できます。

以下に C++ による実装例を示します。

https://beta.atcoder.jp/contests/code-festival-2018-quala/submissions/3147051

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

using namespace std;

int A, B, C;
int S;

int main(){
    scanf("%d%d%d", &A, &B, &C);
    scanf("%d", &S);
    if(A + B + C <= S && S <= A + B + C + 3){
        printf("Yes\n");
    }else{
        printf("No\n");
    }
    return 0;
}
```

B: みかん

長さ N の配列を持ち、それぞれのみかんについて房の個数が A であると明示的に指定されたかどうか管理すればよいです。

以下に C++ による実装例を示します。

https://beta.atcoder.jp/contests/code-festival-2018-quala/submissions/3147011

```
#include<cstdio>

using namespace std;

const int MAX_M = 100;
const int MAX_N = 100;

int N;
int A, B;
int M;
int L[MAX_M], R[MAX_M];

bool flg[MAX_N];

int main(){
    scanf("%d%d", &N, &M);
    scanf("%d%d", &A, &B);
    for(int i = 0; i < M; ++i){
        scanf("%d%d", L + i, R + i);
        L[i]--;
        R[i]--;
    }
    for(int i = 0; i < M; ++i){
        for(int j = L[i]; j <= R[i]; ++j){
            flg[j] = true;
        }
    }
    int ans = 0;
    for(int i = 0; i < N; ++i){
        if(flg[i]) ans += A;
        else ans += B;
    }
```

```
    printf("%d\n", ans);
    return 0;
}
```

C: 半分

各 i について、操作のあとの A_i の値は添字 i に対して施された操作の回数だけに依存します。さらに、cnt_i を A_i の値を 0 にするために必要な操作の回数とするとき、以下のことが成り立ちます。

- 添字 i に対して施された操作の回数が $0, 1, \ldots, cnt_i$ のときの操作後の A_i の値はそれぞれ異なる。
- 添字 i に対して施された操作の回数が cnt_i 以上のときの操作後の A_i の値はすべて等しい。

$cnt_i = O(\log A_i)$ であり、$A_i \leq 10^{18}$ のとき $cnt_i \leq 60$ となります。
以上を踏まえて、この問題を動的計画法で解くことを考えます。$dp_{i,j,k}$ を以下のように定めます。

- $dp_{i,j,0}$ は、$A_1, A_2, \ldots, A_i$ に対して、どの A_l も 0 にならないように操作をちょうど j 回施す方法の数とする。
- $dp_{i,j,1}$ は、$A_1, A_2, \ldots, A_i$ に対して以下の条件を満たすようにちょうど j 回の操作を施す方法の数とする。
  - 操作後においてある A_l の値が 0 となる。
  - どの l についても cnt_l 回以下の操作しか施さない。

この配列の値が求められれば、答えの値 ans は以下のように求められます。

$$ans = dp_{N,K,0} + \sum_{k \leq K} dp_{N,k,1}$$

$cnt_i \leq 60$ であったので、$j > 60N$ のとき $dp_{i,j,k} = 0$ となります。したがって、$j \leq 60N$ について dp の値を求めれば良いことになります。
この範囲で、配列 dp は以下の漸化式に従って計算できます。

$$dp_{i,j,0} = \sum_{j - cnt_i + 1 \leq j' \leq j} dp_{i-1,j',0}$$
$$dp_{i,j,1} = \sum_{j - cnt_i \leq j' \leq j} dp_{i-1,j',1} + dp_{i,j-cnt_i,0}$$

ただし $j < 0$ のとき任意の i, k について $dp_{i,j,k} = 0$ としています。
上の漸化式に従って dp のを計算すると、各値の計算に $O(\log(\max A_i))$ 時間かかるので、全体の計算量は $O(N^2(\log(\max A_i)^2))$ となり、十分高速に答えが求められます。なお、全体で $O(N^2 \log(\max A_i))$ 時間で計算する方法もあります。

実装例: `https://beta.atcoder.jp/contests/code-festival-2018-quala/submissions/3225771`

D: 通勤

$X_0 = 0$ とし、点 X_0 にもガソリンスタンドがあるものと考えます。また、各 i について、点 X_i にあるガソリンスタンドをガソリンスタンド i と名付けます。

この問題は動的計画法で解くことができます。dp_i の値を、ガソリンスタンド i で燃料を補給することになるような、ガソリンスタンド $1, 2, \ldots, i-1$ の建て替え方の数と定めます。

各 i について、j_i を $X_{j_i} > X_i + (F - T)$ を満たす最小の添字と定めます。同様に、k_i を $X_{k_i} > X_i + F$ を満たす最小の添字と定めます[*1]。

このとき、dp_i の値は、$l = j_i, j_i + 1, \ldots, k_i - 1$ に対する dp_l の値に、$2^{j_i - i - 1} dp_i$ だけ寄与します。このことを用いると、適当な累積和をとることにより $dp_0, dp_1, \ldots$ の値が $O(N)$ 時間で求められます。

各 i に対する dp_i の値が求まれば、もとの問題の答え ans は以下のように計算できます。

$$ans = \sum_{D - X_i \leq F} 2^{N-i} dp_i$$

実装例: `https://beta.atcoder.jp/contests/code-festival-2018-quala/submissions/3261392`

[*1] このような添字が存在しない場合の処理の方法の説明は省略しますが、実装の際は注意してください

E: オレンジとみかん

それぞれの人が受け取る果物の個数を S とします。すなわち、$S = (X + Y)/N$ です。

この問題は答えの値で二分探索できる形になっています。よって、以下の問題が解ければ良いことになります。

問題 1　最大の満足度の人と最小の満足度の人の満足度の差が d 以下になるような果物の分け方は存在するか？

このままでは考えづらいので、さらにパラメータを増やした以下の問題をまず考えます。

問題 2　最小の満足度の人の満足度が l であり、最大の満足度の人の満足度が $l + d$ 以下になるような果物の分け方は存在するか？

この条件はさらに、「すべての人の満足度が l 以上 $l + d$ 以下である」と言い換えられます。

この問題は次のようにして解くことができます。すなわち、まず各 i について、$l \leq A_i x + B_i(S - x) \leq l + d$ となる整数 $0 \leq x \leq S$ の最小値 lb_i と最大値 ub_i を求めます。ただし、このような整数が存在しない場合、$lb_i = ub_i = -1$ としておきます。このとき、条件を満たす果物の分け方が存在する条件は、$lb_i = -1$ となる i が存在せず、かつ lb, ub が

$$\sum_{1 \leq i \leq N} lb_i \leq X \leq \sum_{1 \leq i \leq N} ub_i$$

を満たすことです。

これで問題 2 が解けました。すべての l について問題 2 を解くことで問題 1 も解くことができますが、これでは時間がかかりすぎるので高速化することを考えましょう。

l, d の値を決めたときの $lb_i = -1$ となる i の個数、lb_i の和および ub_i の和をそれぞれ $bad(l, d), lsum(l, d), usum(l, d)$ と書くことにします。

d を固定し、l を $0, 1, \ldots$ の順に変化させていったとき、$bad(l, d), lsum(l, d), usum(l, d)$ の値が変化する点はある i および $0 \leq x \leq S$ について以下のいずれかの式が成り立つような点です。

$$l = A_i x + B_i(S - x)$$
$$l + d = A_i x + B_i(S - x)$$

このような点は $O(X + Y)$ 個しかありません。さらに、これらの点の前後での $bad, lsum, usum$ の値の変化も簡単に計算できます。

よって、上記の点を列挙し、小さい順に試していくことで問題 1 を解くことができます。

問題 1 を解くことができたので、これを $O(\log U)$ 回 (U は答えの上限) 解くことでもとの問題を解くことができます。

実装例: `https://beta.atcoder.jp/contests/code-festival-2018-quala/submissions/3261395`