

ABC 143 解説

gazelle, kort0n, beet

2019 年 10 月 19 日

For International Readers: English editorial will be published in a few days.

A: Curtain

答えは、 $A > 2B$ のときは $A - 2B$ であり、 $A \leq 2B$ のときは 0 です。
 A, B を入力として受け取り、答えを計算し、それを出力することにより、AC となります。
尚、答えを計算する際には if 文を用いても構いませんが、max 関数を用いるとより簡潔に書くことが出来ます。
以下は C++ での実装例です。

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     int A, B, ans;
5     cin >> A >> B;
6     ans = max(0, A - 2 * B);
7     cout << ans << endl;
8     return 0;
9 }
```

B: TAKOYAKI FESTIVAL 2019

N が大きくないので、素直な全探索解で正答できます。

つまり、食べるたこ焼き 2 個の選び方を全探索して、それぞれの体力回復量の和をとればいいです。

この問題のように「順番を考慮しない 2 要素の選び方」を全探索する際のポイントとして、下記コードにあるように、内側のループ変数の始点を外側のループ変数 +1 から始めるといいです。こうすることで、内側のループで選ぶインデックスが必ず外側のループで選ぶインデックスより大きくなり、同じ選び方を 2 回見してしまうことを回避できます。

以上の全探索解の計算量は $O(N^2)$ ですが、この問題は工夫すると $O(N)$ で解くこともできます。考えてみてください。

Listing 1 C++ による実装例

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int n;
6     cin >> n;
7     vector<int> d(n);
8     for(int i = 0; i < n; i++) cin >> d[i];
9     int ans = 0;
10    for(int i = 0; i < n; i++) {
11        for(int j = i + 1; j < n; j++) {
12            ans += d[i] * d[j];
13        }
14    }
15    cout << ans << endl;
16    return 0;
17 }
```

C: Slimes

解法 1

同じ文字が連続する S の極大な連続部分文字列を，それぞれグループと見なします．



図 1 $S = aaabcbbbdd$ に対するグループ分けの例

このとき， S の各文字間の境界であって，その両隣の文字が異なるものと，前述したグループの境界は，一対一に対応します．

以上より，前述した S 中の境界に着目することで，解を出力することが出来ます．

C++ による解答例:<https://atcoder.jp/contests/abc143/submissions/8034709>

解法 2

一部の言語では，文字列中の隣り合う同じ要素を，1 つのみ残して削除する関数があります．例えば，C++ では以下のようなコードを書けば良いです．

C++ による解答例:<https://atcoder.jp/contests/abc143/submissions/8034663>

D: Triangles

全ての棒の組について三角形が作れるかをチェックすると、時間計算量が $O(N^3)$ となり、間に合いません。

三角形を構成する棒のうち、1 番目と 2 番目に長いものを固定します (ただし、同じ長さを持つ棒が複数存在する場合は、予め適当に大小関係を定めておきます)。

このとき、3 番目に長い棒として使える棒は、「2 番目の棒より短く、一定以上の長さを持つもの」です。

このような棒の数は、予め棒の長さをソートしておくと、二分探索で効率的に求めることができます。以上より、1 番目に長い棒と 2 番目に長い棒の選び方について探索すると、時間計算量 $O(N^2 \log N)$ で答えを求めることができます。

C++ による解答例:<https://atcoder.jp/contests/abc143/submissions/8034764>

E: Travel by Car

ワーシャルフロイド法というアルゴリズムを用いると、 $O(N^3)$ で全点間の最短距離を求めることができます。

まず、各町を頂点とし、各道を辺とした重み付き無向グラフを作ります。各辺の距離は、問題における道の距離とします。

このグラフにワーシャルフロイド法を適用することで、各町間の最短距離を求めることができます。

次に、もう 1 つグラフを作ります。各町を頂点とし、最短距離が L 以下である町の組の間に、距離 1 の辺を張ります。

このグラフにワーシャルフロイド法を適用することで、各町間を移動する際に必要な燃料補給回数の最小値が求められます。

あとは、この結果に基づいて各クエリに答えれば良いです。時間計算量は $O(N^3 + Q)$ です。

C++ による解答例:<https://atcoder.jp/contests/abc143/submissions/8034843>

F: Distinct Numbers

与えられた数列から、長さ $K (1 \leq K \leq N)$ の狭義単調増加数列を最大何個取り出すことができるかという問題です（要素の順番を入れ替えてもよい）。

$X (1 \leq X \leq N)$ 個の列を取り出すときの最大の長さを表す関数 $f(X)$ を考えます。

これは、 C_j を $A_i = j$ なる i の個数として、

$$f(X) = \left\lfloor \frac{\sum_{j=1}^N \min(C_j, X)}{X} \right\rfloor$$

という式で表せます。また、 D_k を $C_j = k$ なる j の個数とすると、

$$f(X) = \left\lfloor \frac{\sum_{j=1}^N \min(C_j, X)}{X} \right\rfloor = \left\lfloor \frac{\sum_{k=0}^N \min(k, X) \times D_k}{X} \right\rfloor = \left\lfloor \frac{\sum_{k=0}^X k D_k + X \sum_{k=X+1}^N D_k}{X} \right\rfloor$$

と変形できます。したがって、 $\sum_{k=0}^X k D_k$ と $\sum_{k=X+1}^N D_k$ を累積和を用いて計算することで、全ての $f(X)$ の値を $O(N)$ で求めることができます。その後、各 K について、 $K \leq f(X)$ を満たすような最大の X を求めればよいです（存在しなければ 0）。