JOI/JOIG 2023/2024 春季トレーニング 卓球 (Table Tennis) 解説

解説: 蜂矢 倫久 (Mitsubachi)

Step 0

問題概要

0 問題概要

- N 人の総当たり戦の結果を作る
- 条件「3 すくみの数が M 個」を満たしたいです
- 可能か不可能か判定、可能なら 1 つ作ってください
- 1 つの入力ファイルで Q 個解いてください

0 制約

- $1 \leqq Q$
- $N \leqq 5\,000$
- $M \leqq {}_N C_3$
- N の総和は 5 000 以下

0 小課題

小課題番号	内容	配点 (JOI)	配点 (JOIG)
1 (JOIG のみ)	M = 0	0 点	5 点
2	M ≦ N - 2	5 点	12 点
3 (JOIG のみ)	M ≦ (N-1)^2/4	0 点	15 点
4	N の総和 ≦ 7	4 点	10 点
5	N の総和 ≦ 20	23 点	23 点
6	N の総和 ≦ 150	30 点	19 点
7	N の総和 ≦ 600	15 点	8 点
8	追加の制約なし	23 点	8 点

0 小課題

- 以降の小課題の番号は JOIG に準拠します

0 マルチテストケースの実装

- 1つの入力ファイルでテストケースが複数ある問題はたまにみます(マルチテストケースとか言ったりする)
- 以下みたいに書くと楽です

```
void solve(){
 // ここに処理
}
int main(){
  int t; cin >> t;
  while(t--){
    solve()
  }
}
```

Subtask **1**

$M = 0$

1 考察

- 3 すくみが 0 個
- 強い人がとても勝って、弱い人がとても負けてそう

1 考察

- つまり各人に強さを決めて、その強さの大小で勝敗を決めれば OK

Subtask **2**

$M \leqq N - 2$

2 考察

- 強さの設定はそのままにしたい
- 1 個逆転が起こったとする
- 強さが A 番目に小さい人と B 番目に小さい人の結果を逆転すると 3 すくみが B - A - 1 個できる

2　B を探す

- A = 1 とすると B - 2 = M になるような B を見つけたい
- M ≦ N - 2 なのでそのような B は N 以下
- つまり必ず存在！

Subtask **3**

M ≦ (N-1)^2/4

3 考察

- O(N^2) の式ということは N ごとに何か足しそう
- 係数が 1/4 なので 2 人ずつ何かをしそう？
- 2 人以外と 2 人の間の関係を考える

3 考察

- 2 人を A, B として A は B に勝ったとする
- 後の人は全員 A に勝てた or 負けたとしよう
- B に対しても同様
- 全員 A に勝てた & B に負けたとすると 3 すくみは N - 2 個
- N - 2 人の場合の問題になる

3 考察

- $(N - 1)^2/4 - (N - 2) \leqq (N - 3)^2/4$ なので
- N を 2 小さくした小課題 3 の問題に帰着される！
- 実装上の注意として、 N が 6 以下ぐらいで場合分けが必要
- これをどうするかは次の小課題 4 で

Subtask 4

$N \leqq 7$

4 考察

- 試合の数は ${}_N C_2 \leqq 21$
- bit 全探索ができそう
- 3 すくみの数は愚直に数えて O(N^3)
- O(2^(N/2) N^3) で解けました

4 埋め込み

- この場合ありうる (N, M) の数が少ない
- 全通り手で解いて答えをソースコード中に埋め込む
- 出力 O(N^2) で OK！
- 10 ケースを超えるのを手で解くなら多分他の小課題に手をつけた方がいいと思います

Subtask **5**

N ≦ 20

5 3 すくみの数を数える

- 制約に N ≦ 5 000 とある
- 3 すくみの数を O(N^3) より速くジャッジしてそう
- O(N^2) で数える方法を考える
- 3 すくみでない数を数えられないか

5 3 すくみの数を数える

- A, B, C が 3 すくみでないときその 3 人の中の試合結果を考える
- A が B, C に勝って、 B が C に勝ってる感じ
- つまり A が 2 勝している
- N - 1 人のうち A が勝った人のうち 2 人選ぶと、 A が 2 勝してるタイプの 3 すくみになる
- これで O(N^2) で数えられる

5 問題の言い換え

- つまり人 i の勝利数 d(i) を適切に決めて ${}_{d(i)}C_2$ の総和を ${}_NC_3 - M$ にしたい

- まず d(i) の条件を考える $d(1) \leqq d(2) \leqq \ldots$ としよう

- 必要条件を考える

- $0 \leqq d(i) \leqq N - 1$
- d(1) から d(i) までの総和が ${}_iC_2$ 以上
- d(1) から d(N) までの総和が ${}_NC_2$

5 DP

- 逆にこれが十分条件になっている(ランダウの定理)
- 証明: 略
- とにかくこれで DP をできそう
- DP[d(i)まで決めた][ここまでの d(i) の総和][ここまでの ${}_{d(i)}C_2$ の総和]
- 計算量 O(N^8) で d(i) を求められる

5 結果を作る

- d(i) から試合結果を作りたい
- 人 N に着目 この人は d(N) 勝している
- 直感: 人 1 から人 d(N) までには勝ってそう
- これでやると実はうまくいく
- 証明: ランダウの定理の正当性を認めるとできます

5 結果を作る

- 人 N は人 1 から人 d(N) までには勝って人 d(N) + 1 から人 N - 1 までには負けたとする
- d(d(N) + 1) から d(N - 1) までを 1 減らす(人 N に勝ったので)
- d(1) から d(N - 1) を sort する(これで人の順番が変わるので最初どの人だったかを管理する)
- O(NlogN) で人を 1 人減らせるので O(N^2logN)
- N - 1 人の結果をもらった後 N 人の場合の番号に直すと O(N^3)

5 解法

- DP で d(i) を求める O(N^8)
- 再帰的に試合結果を作る O(N^2logN) or O(N^3)
- 全体で O(N^8) で解けました

Subtask **6**

$N \leqq 150$

6 判定問題を考える

- d(i) を作るところを速くしたい
- ところで Yes か No の判定は？
- ${}_{d(i)}C_2$ の総和を ${}_{N}C_3$ - M にできるか
- 最大値は M = 0 のとき これは小課題 1 で達成してる
- 最小値を求めよう

6 最小値を求める

- 一旦条件「$d(1)$ から $d(i)$ までの総和が ${}_iC_2$ 以上」を無視して「$0 \leqq d(i) \leqq N - 1$」「$d(1)$ から $d(N)$ までの総和が ${}_NC_2$」だけ考える

- このとき $i < j$ について $d(i) + 2 \leqq d(j)$ なら $d(i)$ に 1 を足して $d(j)$ に 1 を引くと ${}_{d(i)}C_2$ の総和が小さくなる
- $({}_{d(j)}C_2 + {}_{d(i)}C_2) - ({}_{d(j)-1}C_2 + {}_{d(i)+1}C_2) = d(j) - d(i) - 1 \geqq 1$

- そのような i, j がない状態が最小
- つまり $d(1) + 2 > d(N)$

6 最小値を求める

- このような d(i) は N の偶奇で異なる
- N が偶数のとき d(i) は N / 2 と N / 2 - 1 が N / 2 個ずつ
- N が奇数のとき d(i) は全部 (N - 1) / 2
- またこのとき「d(1) から d(i) までの総和が ${}_iC_2$ 以上」が成り立っている
- ${}_{d(i)}C_2$ の総和の最小値が求まりました

6 間はどうか

- ${}_{d(i)}C_2$ の総和が最小となる実際の試合結果は人 i が i + 1 から i + d(i) までに勝って、残り負けてるとすれば実現可能
- ${}_{d(i)}C_2$ の総和の最小値と最大値の間にしたいときどうしよう
- $({}_{d(j)}C_2 + {}_{d(i)}C_2) - ({}_{d(j)-1}C_2 + {}_{d(i)+1}C_2) = d(j) - d(i) - 1$ を思い出すと、 $d(i) = d(j)$ $(= x)$ ならこの 2 人の勝利数を $x + 1$ と $x - 1$ にすると 1 増やせる
- これを ${}_{N}C_3$ - M - (${}_{d(i)}C_2$ の総和の最小値) 回すればいい

6　間はどうか

- 実はこれが可能
- i と j の間の試合結果を入れ替えてみる
- i が j に勝ってたのを入れ替えると i は x - 1 勝、 j は x + 1 勝になる 元々 i が j に負けてたら i は x + 1 勝、 j は x - 1 勝に
- いずれにせよ d(i) = x となる 2 人を x + 1 と x - 1 にできる

6 解法

- 1 回あたりの操作は i, j を見つけるのに O(N)
- 操作回数は O(N^3)
- なので O(N^4) でできる
- さっきの方法で直接試合結果を作ってもいいし小課題 5 で述べた方法で d を作ってから試合結果を作ってもいい
- どれでも O(N^4) とかで解けます

Subtask **7**

$N \leqq 600$

7 高速化

- d を変えるところを高速化しよう

- 案 1: 1 回あたりの操作を O(N) より速くしたい
- x 勝した人の番号とかを管理して、 x 勝した人が 2 人以上いるような x の一覧を set とかで管理すれば O(logN) などで OK
- もうちょっと頑張ると O(1) とかにもできます

7 高速化

- 案 2: 操作回数を O(N^3) より小さくしたい
- d(N) = N - 1 とする このとき人 N がいなくとも他の d は変わらない
- ${}_{N}C_{3}$ - M - ${}_{N-1}C_{2}$ が N - 1 のときの ${}_{d(i)}C_{2}$ の総和の範囲なら N が小さい問題になる
- 逆にこれができないとき操作回数は O(N^2) になってる

- 案 1 と案 2 のどちらを採用しても O(N^3) とか O(N^3logN) で解けます

Subtask **8**

$N \leqq 5\,000$

8 解法

- 案 1 と案 2 の両方を採用すれば O(N^2) とか O(N^2logN) で解けます

- d から試合結果を作るのを O(N^3) だと難しいかも
- 実は O(N^2) で作るアルゴリズムがあったります

Step 9

補足

9 違う視点で見る

- 乱択や山登りができそう
- 実際書いてみると N ≦ 150 ぐらいまでは行けたりする
- 何も思いつかなかったら乱択や山登りを書いてみるのもあり

Step **10**

得点分布

10 得点分布(JOI)

- 100点：？人
- 77点:？人
- 62点:？人
- 27点:？人
- 9点:？人
- 5点:？人
- 4点:？人

10 得点分布(JOI)

- 100点：1人
- 77点: 1人
- 62点: 3人
- 27点: 1人
- 9点: 11人
- 5点: 3人
- 4点: 1人

10

得点分布(JOIG)

- 42点 : ?人
- 17点: ?人
- 15点: ?人
- 5点: ?人

10 得点分布(JOIG)

- 42点 : 1人
- 17点: 2人
- 15点: 1人
- 5点: 4人