二次予選 3
イベント巡り 解説

問題概要

- N個のイベントがありパラメーターP_i, S_iを持つ
- これらのイベントの中から、以下の条件を満たすイベントの集合の最大の大きさを求める

すべての$S_i < S_j, P_i \neq P_j$を満たすイベントの組(i, j)について

$$S_i + 1 + D + |\{S_x \leq S_i\}| \times K \leq S_j$$

を満たす

小技

- 隣町のインデックスを取り出したいときは

$$3 - p, p\text{^}3$$

とすると実装が楽です

- (0-indexedの場合は$1 - p, p\text{^}1$)

- この解説では隣町を表すのは$3 - p$としています

小課題 1

- $K = 0, N \leq 20$

- 全探索を行う
- それぞれのイベントについて参加するかしないかを割り当て、実行可能か判断する

- $O(N\ 2^N)$

小課題 2

- $K = 0, N \leq 4000$
- DPを行う
- イベントiに参加するときの、時刻$S_i + 1$までのイベント参加回数の最大値を$dp[i]$とする

小課題 2

- イベントaに参加した後にイベントbに参加できることを$a \to b$と表記することにする

- 更新式は

$$dp[i] = \max\{dp[j] \mid j \to i\} + 1$$

- $O(N^2)$

小課題 3

- $K = 0$
- 小課題 2 の解法を高速化することを考える
- 事前にSによってイベントをソートしておき、$S_i < S_{i+1}$となるようにしておく

小課題 3

- 時刻$S_i + 1$に町pにいるときの、時刻$S_i + 1$までのイベント参加回数の最大値を$dp'[i][p]$とする

- 更新式は、隣町から移動してきた場合の項を

$$m_s = \max\left\{dp'[j][3-p] \middle| S_j + 1 + D \leq s\right\}$$

とすると

$$dp'[i][p] = \max\left(dp'[i-1][p], m_{S_i}\right) + 1 \; (P_i = p\text{のとき})$$
$$dp'[i][p] = \max\left(dp'[i-1][p], m_{S_{i+1}}\right) \quad (P_i \neq p\text{のとき})$$

小課題 3

- m_sを高速に求めることができればこの小課題を解くことができる

- 二分探索によりm_sを高速に求めることができる
  - S_jは前処理によって単調増加となっている
  - $dp'[j][p]$はjを動かしたときに単調増加である

- $O(N \log N)$

小課題 4

- $N \leq 160$
- DPを行う
- イベントiに参加するとき、時刻$S_i + 1$までのイベント参加回数をxとすることが可能ならば$dp''[i][x] = True$、それ以外は$dp''[i][x] = False$とする

小課題 4

- 時刻$S_a + 1$までに参加したイベントの数がxであるとき、イベントaに参加した後にイベントbに参加できることを$a \rightarrow_x b$と表記することにする
- 更新式は

$$dp''[i][x] = \bigvee_{j \mid j \rightarrow_{x-1} i} dp''[j][x-1]$$

- $O(N^3)$

小課題 5

- $N \leq 4000$
- 小課題 4 を高速化したい
- $dp''[i][x] = True$であるとき、$dp''[i][x-1] = True$であることがわかる
- そこで各iについて$dp''[i][x] = True$となる最大のxを考え、それを$dp[i]$とする

小課題 5

- イベントiに参加してからイベントjに参加するとき、時刻S_j+1でのイベント参加数の最大値を$M(i,j)$とする
  - このとき、$M(i,j)$は$dp[i]$から定数時間で求められる

- 更新式は

$$dp[i] = \max(M(j,i) \mid j \to_1 i)$$

- $O(N^2)$

小課題 6

- 満点解法

- 小課題 5 をさらに高速化したい

- 事前にSによってイベントをソートしておき、
  $S_i < S_{i+1}$となるようにしておく

小課題 6

- 考察として、$M(i, j) = dp[i] + 1$となるもののみを考慮してもDPテーブルは同じものとなる

- すなわち、更新式は
$$dp[i] = \max(dp[j] + 1 | \ j \rightarrow_{dp[j]} i)$$
となる。

- ここから小課題 3 のようにDPテーブルを変える

小課題 6

- 時刻$S_i + 1$に町pにいるときの、時刻$S_i + 1$までのイベント参加回数の最大値を$dp'[i][p]$とする
- 更新式は、隣町から移動してきた場合の項を

$m_s = \max\left\{dp'[j][3-p] \middle| S_j + 1 + D + dp'[j][3-p] \times K \leq s\right\}$

とすると

$$dp'[i][p] = \max\left(dp'[i-1][p], m_{S_i}\right) + 1 \ (P_i = p\text{のとき})$$
$$dp'[i][p] = \max\left(dp'[i-1][p], m_{S_i+1}\right) \quad (P_i \neq p\text{のとき})$$

小課題 6

- 解法その 1
- m_sを高速に求めることができればこの小課題を解くことができる
- 二分探索によりm_sを高速に求めることができる
  - $S_j + 1 + D + dp'[j][3-p] \times K$は$j$を動かしたとき単調増加となっている
  - $dp'[j][p]$はjを動かしたときに単調増加である
- $O(N \log N)$

小課題 6

- 解法その 2
- m_sを高速に求めるために尺取り法を使うこともできる
- $O(N\log N)$