Treatment Project

解説

nxteru

概要

- コストを使って、ある日にある連続した区間を治療する計画が与えられる
- 治療された区間は日が経つにつれて縮んでいく
- 全体を治療するのにかかるコストの最小値を求めよ

小課題1（4点）

- T_i＝1
- コスト付きの区間が与えられるので、全体を覆うコストの最小値を求める問題

小課題1（4点）

- 計画をRでソート
- dp[i]…i番目までの計画を使いR[i]まで覆うようにするときのコストの最小値とする
- dp[i]=min{dp[j]｜j<iかつL[i]-1<=R[j]}+C[i]
- このjは連続した範囲になっているので最小値を求めるsegment treeで高速化できる
- O(MlogM)

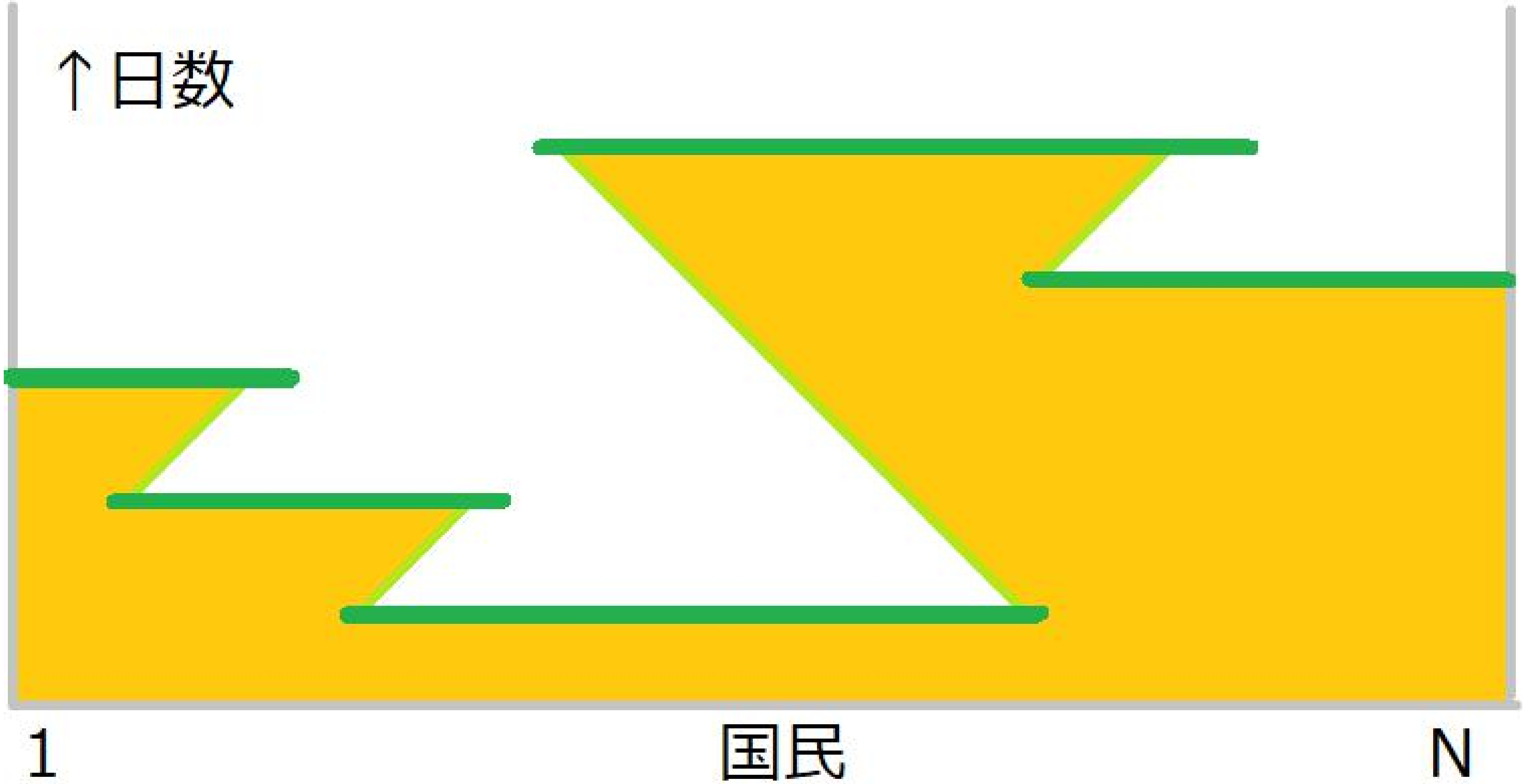
小課題2（5点）

- M<=16
- 2^M通りの計画の採用の仕方をすべて調べることができる
- 採用した計画で条件を満たしているかの判定が少し煩雑
- Mが小さいので少しさぼっても大丈夫
- 自分の実装だとO(2^MM^2)

小課題3（30点）

- M<=5000
- ここからちゃんと考えないといけない
- 小課題2でめんどくさかった条件を満たすかの判定について考える
- 条件を満たす計画の採用の仕方はどのような計画の集合だろう？

条件を満たす計画の集合の例



考察

・治療された区間が狭まりながら計画どうしがつながって感染した領域と治療された領域を分ける境界を形成している

・つまり国民1を含む計画からつながっている計画をたどって国民Nを含む計画まで行ければよい

→最短経路問題になった

小課題3（30点）

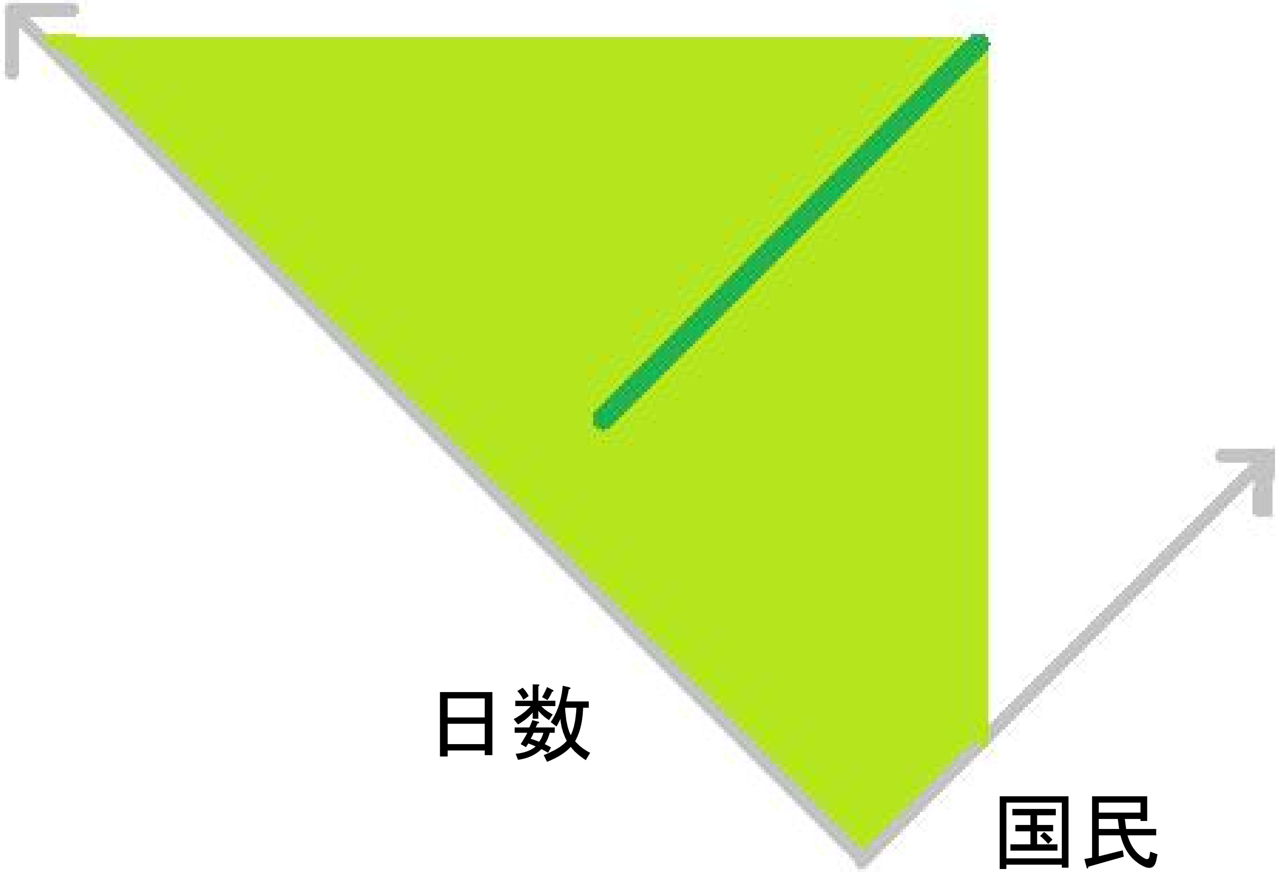
- 計画aから計画bに行けるときaからbにコストC[b]の有向辺をつくる
- その条件はL[b]<=R[a]-|T[b]-T[a]|+1と表せる
- これをすべての計画の組について調べて辺をはり、国民1を含む計画から国民Nを含む計画までの最短距離を求める
- ダイクストラ法などでO(M^2)

小課題4（61点）

- 追加の制約なし(M<=100000)
- 辺をすべてはることもできない
- 小課題3では最短経路問題をそのまま解いただけ
- 今まで使っていなかったこの問題特有の性質を考える

いい性質①

・45度回転させるとある計画(下図の緑)から辺がでる計画は下図のような長方形領域に左端がある計画になる



X[i]=L[i]-T[i],Y[i]=L[i]+T[i]
とすると
aからbに遷移できるのは
X[b]<=R[a]-T[a]+1
かつ
Y[b]<=R[a]+T[a]+1
のとき

いい性質②

・ある計画aに入ってくる辺のコストはすべてC[a]で辺の始点に関わらず同じ
・問題を解くときにダイクストラ法を使う
・するとある計画aの距離が最初に更新された段階でそれが最短距離になっている
→一度距離を更新したらその計画はもう更新しなくていい！

小課題4（61点）

- この2つの性質を使ってpriority_queueを使ったダイクストラ法をする
- 距離の更新をするときに長方形領域の点を見て更新しそれ以降見ないようにする
- 必要なのは長方形領域の点を列挙して削除するデータ構造

小課題4（61点）

- 計画をXでソート、その状態からYの値でマージソートする過程をseg木に保存
- seg木の葉ノードは1つの計画を持っていて葉から順に2つの子ノードをYの値でマージしていく
- 各ノードはXの対応する範囲の計画を、Yでソートした列を持っている
- O(MlogM)で構築できる

小課題4（61点）

・ダイクストラ法を行い、ある計画aが取り出されたらseg木のX<=R[a]-T[a]+1に対応するノードにアクセスする

・アクセスしたノードではYの小さい順に見てY<=R[a]+T[a]+1を満たす間、その計画を取り出していく(取り出したらseg木から削除する)

→更新するべき計画の列挙、削除ができた

小課題4（61点）

・priqueueに同じ計画が2度入らないようにすると
・priqueueの計算量はO(MlogM)
・seg木のノードへのアクセスは合計O(MlogM)
・seg木内の計画はO(MlogM)個あるが同じ計画が2回以上見られることはないので合計O(MlogM)

→満点!!

得点分布