Treatment Project

Explanation

nxteru

overview

- Use costs to create a continuous section on a certain day. be given a plan to treat
- The treated section will shrink as days pass. •Find the minimum cost to treat the entire area.

Meyo

Small topic 1 (4 points)

- T_i=1
- Since an interval with a cost is given, the problem is to find the minimum value of the cost that covers the whole.

Small topic 1 (4 points)

- Sort plans by R
- dp[i]...Use the i-th plan to cover up to R[i]
  - dp[i]=min{dp[j] | j<i and L[i]-1<=R[j]}

+C[i] •This j is continuous Since it is a range, you can speed up the process by using a segment tree to find the minimum value.

- O(MlogM)

Small topic 2 (5 points)

- M<=16
- You can check all 2^M ways to adopt a plan. •You can judge whether the adopted plan satisfies the conditions.

A little complicated

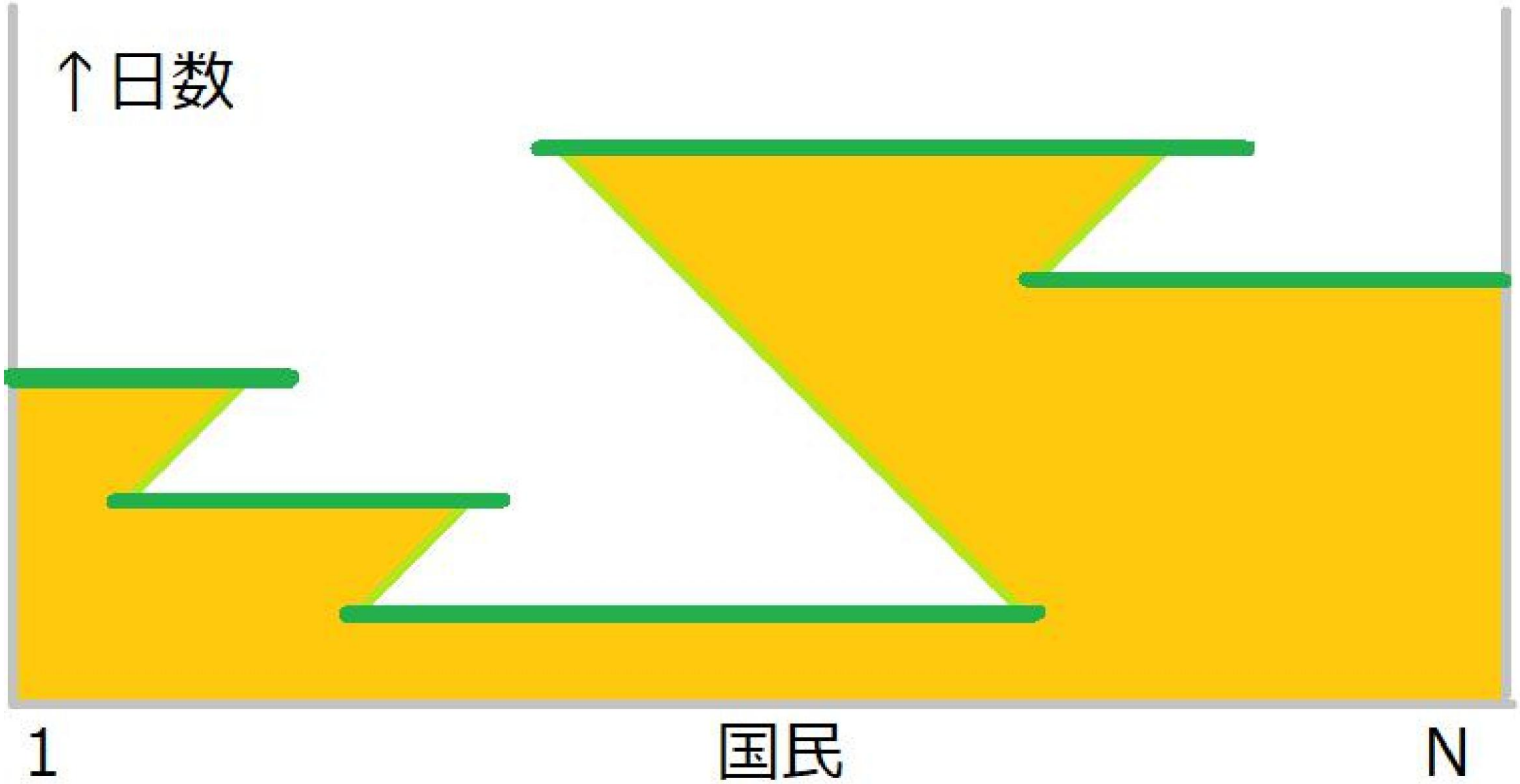
- Since M is small, it is okay to skip a little. •In my implementation, it is O(2^MM^2)

Small topic 3 (30 points)

- M<=5000
- You have to think carefully from here •Judging whether the troublesome conditions in sub-task 2 are met think about the
- What kind of plan should be used to adopt a plan that meets the conditions?

  Is it a collection of paintings?

Example of a set of plans that satisfy the conditions



inspection

•As the treated area narrows, the plans are connected to form a boundary that separates the infected area from the treated area. •In other words, follow the connected plans from the plan that includes Nation 1 to Nation N. As long as you can make a plan that includes

ÿIt became a shortest path problem.

Small topic 3 (30 points)

•When you can go from plan A to plan B, the cost from a to B is Create a directed edge of C[b] •The condition can be expressed as L[b]<=R[a]-|T[b]-T[a]| +1 •Check this for all pairs of plans Find the shortest distance from the plan that includes citizen 1 to the plan that includes citizen N •O(M^2) using Dijkstra's method etc.

Small topic 4 (61 points)

•No additional constraints (M<=100000)

•It is not possible to scale all the edges •In

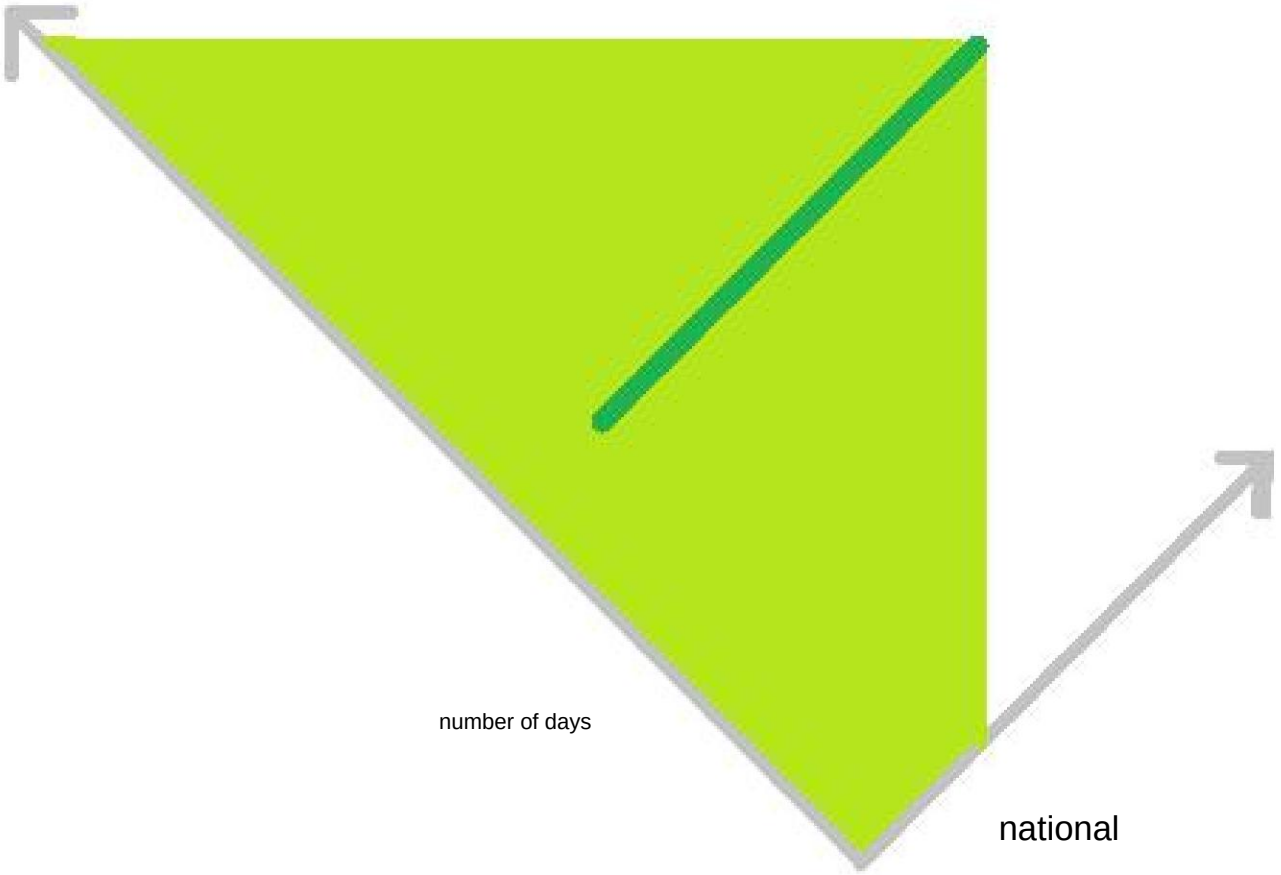
sub-task 3, you were able to solve the shortest path problem as is •I

learned the unique properties of this problem that I had not used before

think

Good property ÿ :

When rotated by 45 degrees, a plan in which edges appear from a certain plan (green in the figure below) becomes a plan with the left edge in a rectangular area as shown in the figure belo



If X[i]=L[i]-T[i],Y[i]=L[i] +T[i],
the transition from a to b is
X[b]<=R[a]-T[a]+1
and
When Y[b]<=R[a] +T[a]+1

Good qualitiesÿ

•All the costs that come into a plan a are C[a]
is the same regardless of the
starting point of the edge •Use Dijkstra's method when
solving the problem •Then, when the distance of a certain plan A
is first updated, it becomes the shortest
distance ÿ Once the distance is updated, the plan becomes No
need to update anymore!

Small topic 4 (61 points)

- Use these two properties to perform Dijkstra's algorithm using priority_queue
- When updating the distance, look at the points in the rectangular area Update it and don't watch it again
- What is needed is a data structure that enumerates and deletes points in a rectangular area.

Small topic 4 (61 points)

- Sort the plans by X, and from that state merge by the value of Y. Save the sorting process in a seg tree. - A leaf node of the seg tree has one plan, and from the leaf, merge two child nodes by the value of Y.•Each node has a column of plans for the corresponding range of X, sorted by Y.

•Can be constructed with O(MlogM)

Small topic 4 (61 points)

- Perform Dijkstra's algorithm, and when a certain plan a is extracted, access the node corresponding to X<=R[a]-T[a]+1 in the seg tree
- For accessed nodes, look in descending order of Y.

  While Y<=R[a]+T[a]+1 is satisfied, extract the plan (after extracting it, delete it from the seg tree)

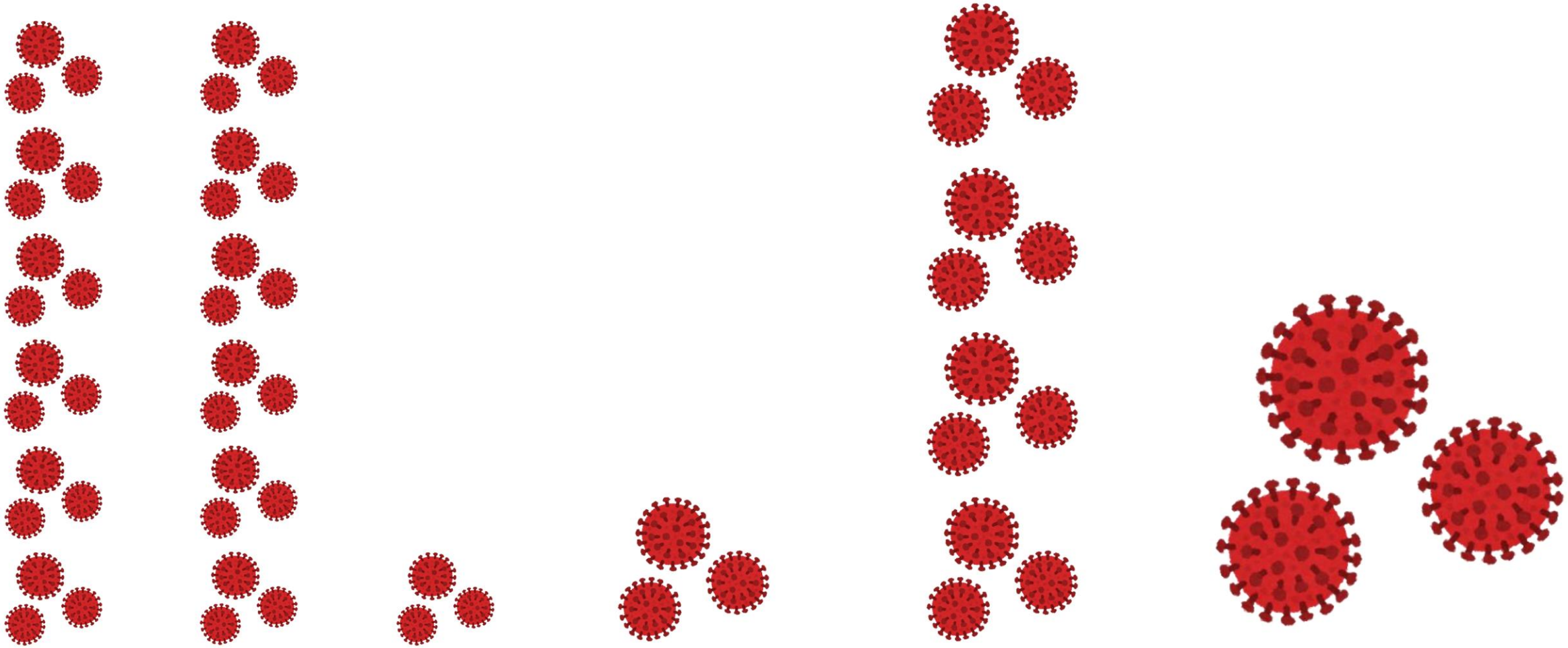
ÿI was able to enumerate and delete plans that should be updated.

Small topic 4 (61 points)

•If you prevent the same plan from entering priqueue twice, •The amount of calculation for priqueue is O(MlogM). •The total number of accesses to nodes in the seg tree is O(MlogM). •There are O(MlogM) plans in the seg tree. The same plan is never seen more than once, so the total is O(MlogM)

ÿFull score!!

Score distribution



0 o'clock 4 o'clock 9 o'clock 35 o'clock 39 o'clock 100 points