



Test Data for Trail Maintenance

Algorithms:

Algorithm 1: After each year, compute the minimum spanning tree (MST). When computing the next year's MST, only consider the trails within the previous year's MST and the trail added.

This algorithm takes $O(N Y \log N)$. It is expected to receive full points.

Algorithm 2: Use a true incremental minimum spanning tree. Each year, determine the path between the endpoints of the new trail and find the maximum length trail in that path. If the length of this maximum trail is greater than the new trail, delete that trail and add the new one. Otherwise, ignore the new trail.

This algorithm takes $O(N Y)$. It is expected to receive full points.

Algorithm 3: Recompute the MST after each year, considering all trails ever seen. Only consider the best trail between any pair of nodes. Use Prim's or Kruskal's algorithm to compute the MST.

This algorithm takes $O(Y^2 \log N)$ time. It is expected to receive around 60% of the points.

Algorithm 4: Recompute the MST after each year, considering all trails ever seen. Use Prim's or Kruskal's algorithm to compute the MST.

This algorithm takes $O(Y^2 \log Y)$ time. It is expected to receive around 50% of the points.



Test Data

Test #	Points	N	W
1	5	10	30
2	5	20	40
3	5	50	70
4	5	80	100
5	5	120	200
6	5	200	300
7	5	200	500
8	5	200	1000
9	5	200	1500
10	5	200	2000
11	5	40	3000
12	5	50	4000
13	5	200	4000
14	5	20	6000
15	5	40	6000
16	5	60	6000
17	5	80	6000
18	5	100	6000
19	5	200	6000
20	5	200	6000