JOI Open 2014 Day1 Factories

overview

- Given a tree with a cost on the edge
- Process a lot of the following queries
  - Given disjoint vertex sets X and Y , find the minimum distance when moving from a point in X to a point in Y.

Small topic 1

ÿ N, Q <= 5000

ÿ Allows O(N) per query

ÿThu DP

ÿ Do DP while returning (the distance from that point to the nearest point in X, Y) and you will get the answer.
let

ÿAlternatively , Dijkstra is O(QN log N)

Small topic 2

ÿEach S, T <= 20

ÿ Allows O(ST) per query

ÿ Just try all point pairs

ÿWith a rooted tree, the distance from the root of each point is calculated, and the LCA can be calculated in O(1)

With some preparation, the distance between two points can be calculated in O(1)

perfect solution

ÿ The sum of S and T is less than 1,000,000 each

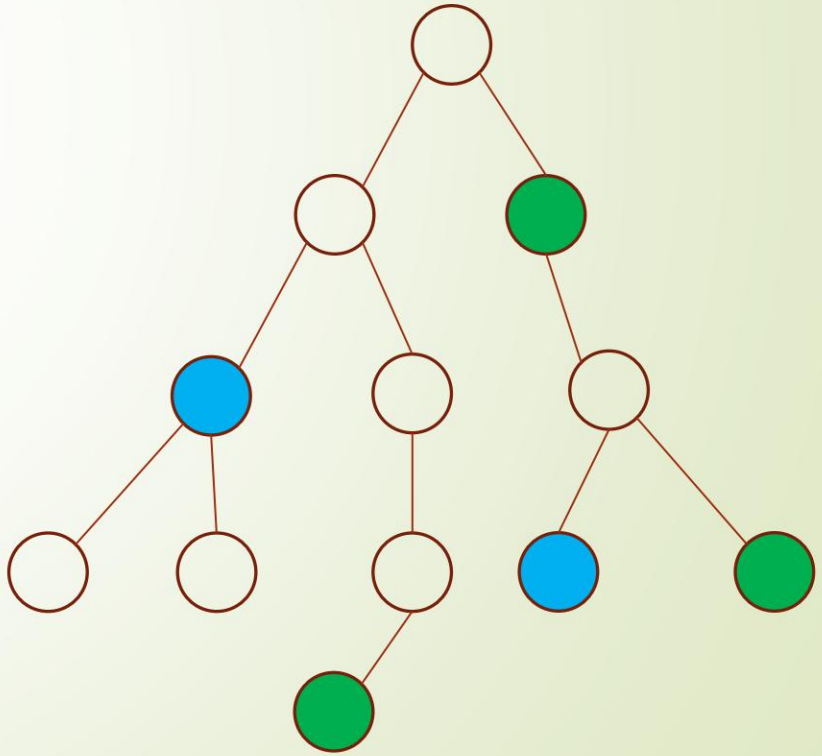
ÿCannot be solved faster than linear in input size

ÿ O(S+T) or O((S+T) log N)

observation

ÿ Do we have to see all N vertices when doing tree DP? ÿ Show

X points in blue and Y points in green



observation

ÿIf you just want to do tree DP, the vertices you need are

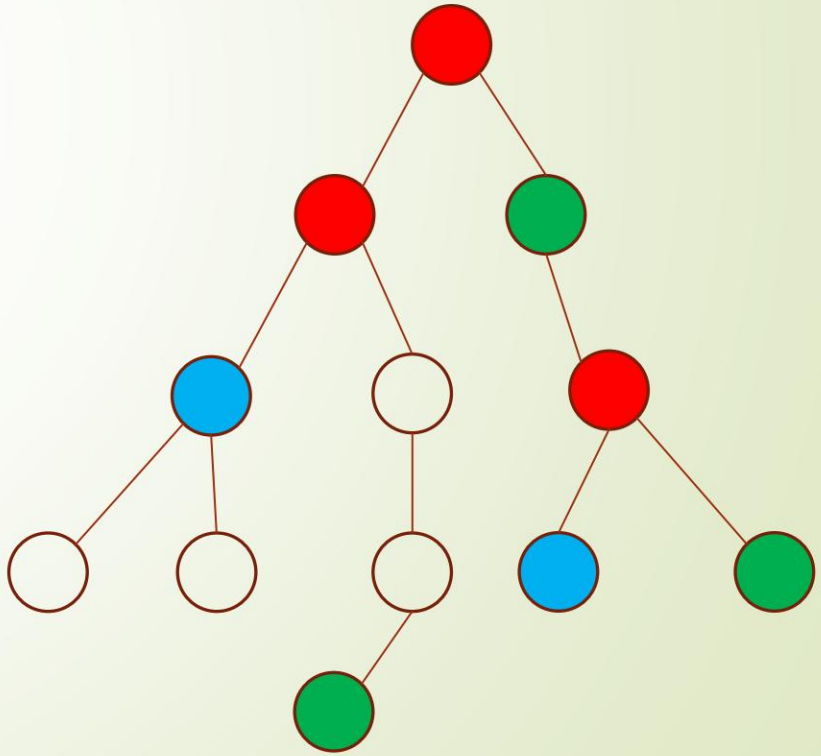
- ÿ Points included in X, Y from the beginning
- ÿ LCA of points included from the beginning

ÿOnly (red vertex in the right figure)

ÿEven if other points are examined,

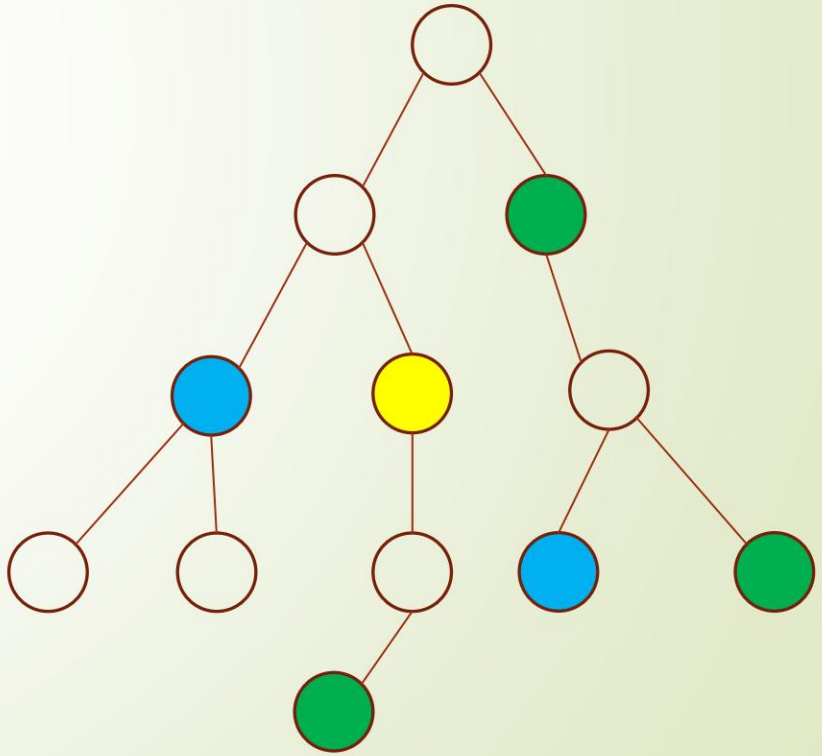
- ÿThere is no point included in X, Y below
- ÿParry values coming from below as they are

Either ÿ , meaningless



observation

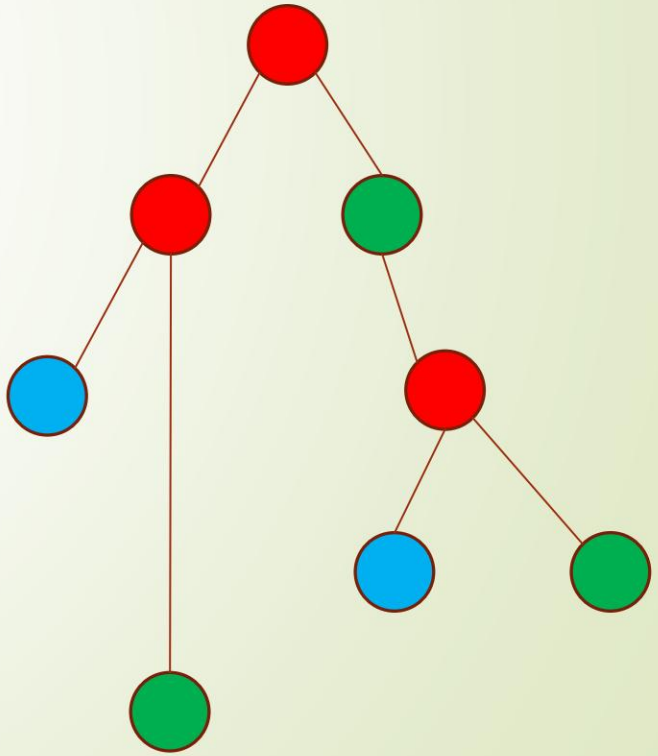
ÿFor example, the yellow point is almost meaningless because it just adds the length of the side to the answer of the point directly below when DP is done.



observation

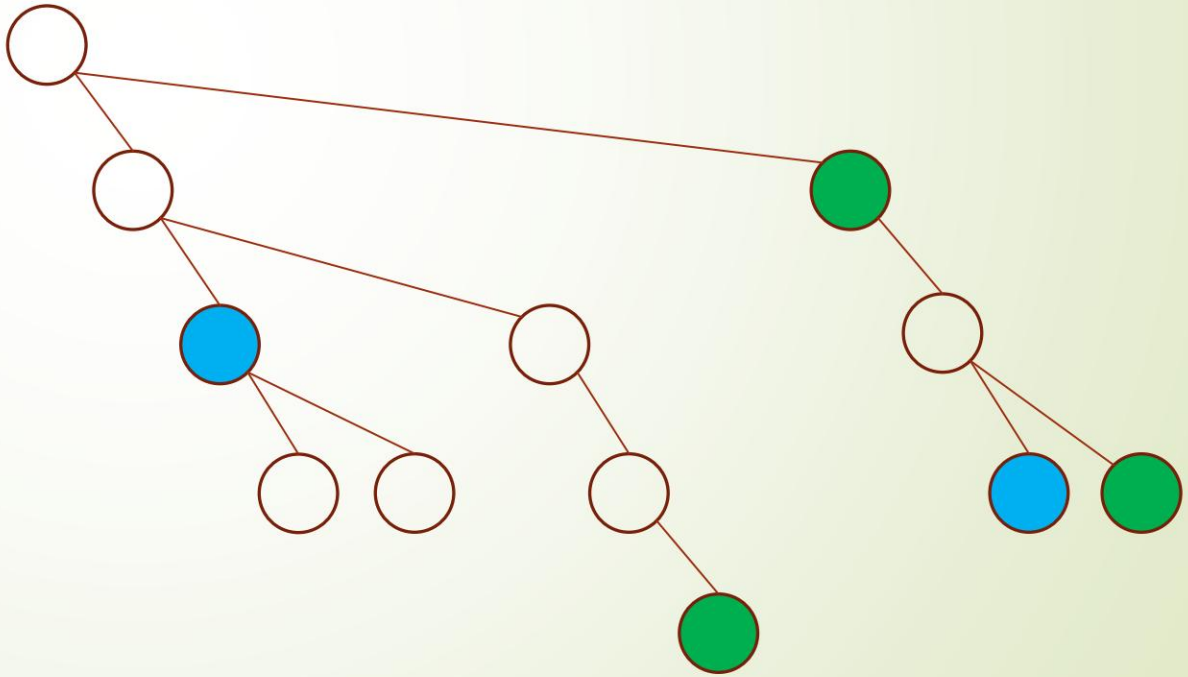
ÿIn the end, you can do DP on the right graph ÿWhen

you collapse the points, just find the sum of the lengths of the sides



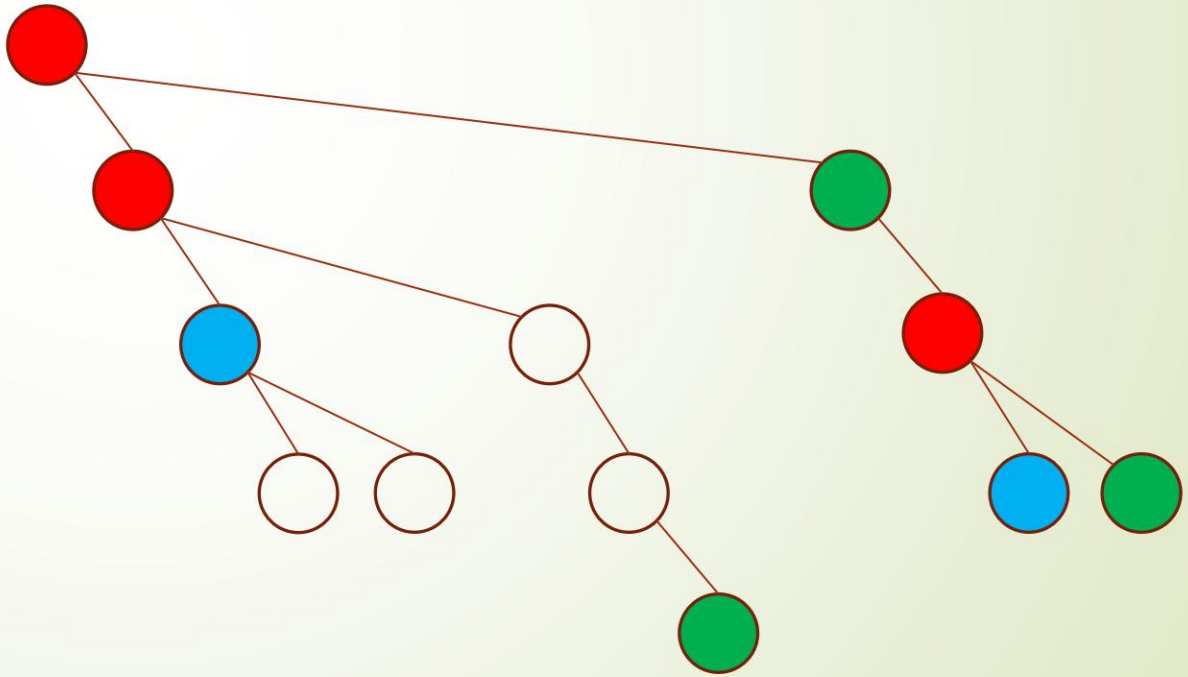
Extract the desired part of the tree

ÿFirst , give the tree a DFS order



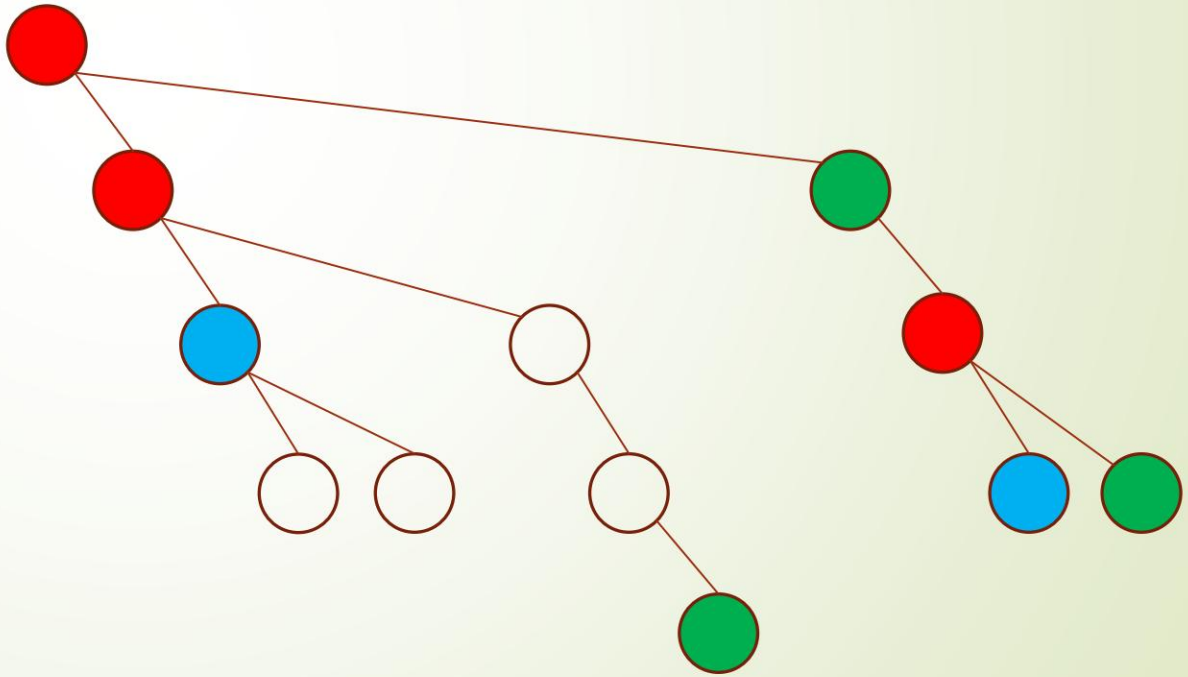
Extract the desired part of the tree

ÿActually , it is enough to see only LCA between adjacent points in this order.



why?

ÿ In this order, the LCA between ``non-adjacent'' points is also
Just say what appears as an LCA



why?

ÿ Consider the LCA of two points P and Q (let it be R)

ÿ Omit if R matches P or Q

ÿSuppose that P = S1 , …, Sk = Q are adjacent in this order on the column

ÿ A child of R that includes Si in its descendants is Si '

other hand, Si-1 'Si' is different, the LCA of Si and Si-1 will be R ÿ (or higher) ÿ Similarly, the

LCA, P' and Q' are different

why?

ÿ From the definition of LCA, P'= S1 ' andQ'= Sk '

are different ÿTherefore , it is impossible that S1 ', …, Sk ' all match

ÿTherefore , there exists two adjacent points on the DFS sequence whose LCA is R.

Evaluation of the number of points

ÿThe number of points initially given is S + T

ÿThe number of newly required points is at most S + T – 1

ÿTherefore , the total number of points required for DP is O(S+T)

ÿOnce you know the structure of the tree including new points, you can do DP in O(S+T)

An example of how to make a "new tree"

ÿFirst , the first points are sorted in DFS order regardless of their X and Y origins.

ÿCalculate LCA between adjacent points

ÿ By using RMQ, etc., it is possible to find which part of the LCA sequence is closest to the root.
to make

An example of how to make a "new tree"

ÿ How to construct a tree over a range of points

ÿ Use RMQ to find where in the LCA sequence the root is in that range ÿ Such points may appear multiple times, but you can choose anywhere

ÿConstruct a tree recursively on the left and right sides of the point

ÿConnect the left and right roots with the current root to end use

ÿThe same point may appear multiple times, but it is useless because it is only connected by zero-length edges

question

Calculations

ÿInitialization _

ÿ O(N) to create DFS tree ÿ O(N) to prepare LCA in O(log N) ÿ Queries (sum of S and T is Ssum and Tsum, respectively) ÿ Sort in DFS order O((S + T) log (S + T)) ÿ O(S + T) to find all LCAs ÿ O((S + T) log (S + T))

ÿIn the end, O((N + Ssum + Tsum) log N) ÿa perfect score