Change log:

- Problem 2 (version 3): redefine crossing point to include starting point, as well as column ids. Also clarified monotonic is non-strict.

- Problem 4 (version 2): specified that the set of $k$-hop edges form a graph, and the goal is to find max weight spanning forest of it.

- Problems 5 and 8 (version 3): specified that the sequence of operations also has length $n$.

This problem set has a total of 9 written problems, plus 1 coding problems, on 3 pages. Written solutions should be submitted to Canvas by 2pm Wednesday Apr 7, 2021. Coding solutions should be submitted separately.

While there are explanations of some of these problems available on the internet. Please refrain from directly copying code you find.

CODE Separate submission on Canvas, 5 points.

https://dmoj.ca/problem/noiwc09p1

Given an 6-by-$10^5$ grid of numbers, answer a sequence of up to $10^6$ operations of the following form in 2 seconds:

(a) Modify the value of some cell.
(b) Find the shortest path, measured in the total values of nodes traversed, between two cells.

note: Time limit is very strict, my guess is only C++/Pascal can get full points.

1. Let the number of rows be $r$, number of columns and number of operations be $n$. Give an algorithm that answers the entire sequence in $O(r^3 n \log n)$ time.

2. Prove the following structural theorem about merging two shortest path tables: consider an $r \times n$ grid of values such that all shortest paths are unique. For some middle column $j$, and rows $i_1$ and $i_2$, define $\text{cross}(j, (i_1, 1), (i_2, n))$ be the row of the last cell in columns 1...$j$ visited by the shortest path from $(i_1, 1)$ to $(i_2, n)$. Prove that for a fixed $i_1$, $\text{cross}(j, (i_1, 1), (i_2, n))$ is non-decreasing as $i_2$ increases.
3. Improve the runtime above to $O(r^2 \log r \cdot n \log n)$. You may find the conclusion from the above problem useful.

For the next two problems, you might want to use the fact that given any tree on $n$ vertices, we can find a vertex whose removal partitions the tree into pieces with at most $(2/3)n$ vertices each.

4. Given a positive weighted undirected tree and a parameter $k \geq 1$, consider the graph formed by connecting each vertex with all vertices whose hop (unweighted) distance in the tree is exactly $k$, with weight equalling to the total weight of the tree path. Show that the max weighted spanning forest of this graph can be computed in $\tilde{O}(n)$ time.

5. Show that a sequence of $n$ of the following operations can be maintained on a weighted undirected tree with $n$ vertices labeled by integers from $1 \ldots n^2$ in $\tilde{O}(n)$ time:
   
   (a) Reweight an edge.
   (b) Relabel a vertex.
   (c) Answer, for a vertex $u$, the minimum distance to some vertex with label in range $[l, r]$.

6. Same as the problem above, but relabel operation changes to ‘add $x$ to labels of all vertices in some subtree (specified via a tree edge and an end point)’, and runtime goal changed to $\tilde{O}(n^{1.9})$. (note: the 1.9 may not be tight, it’s just to allow more shenanigans)

The following problems deal with cactus. Formally, a cactus is an undirected graph where each edge has at most one cycle going through it, and a collection of cactus is known as a desert :-).

7. Two vertices $u$ and $v$ is 3-edge-connected if there are three edge disjoint paths from $u$ to $v$. Show that a graph without a pair of 3-edge-connected vertices is a cactus.

8. Show that a sequence of $n$ of the following sequence of operations can be maintained dynamically on a cactus of size $n$ in $\tilde{O}(n)$ time:
   
   (a) Add/remove/reweight an edge.
   (b) Query for the weight of the shortest path from some $u$ to $v$.  

2
9. Show that given a sequence of $n$ edge insertions/deletions in an undirected unweighted graph, we can answer $n$ queries of the form of ‘between times $[t_1, t_2]$, was $u$ and $v$ ever 3-edge connected’ in $\tilde{O}(n^{1.9})$ time. (note: the 1.9 may not be tight, it’s just there to allow more shenanigans)