- vector / pq are better than map / multiset
- for dp:
 - greedy fails?
 - Hamiltonian Paths
- graph:
 - adj matrix?
 - dijkistra? floyed? bellman-ford?
 - multisource?
 - topo sort? with dp?
 - is DAG?
 - MST? and LCA solution
 - cycles?
 - SCC? and contract components?
 - 2-sat
 - Hamiltonian Paths
 - inverse graph??
 - Bridges and Articulation points?
 - is Bipirtite?
- genenral:
 - stacks for nearest values?
 - tearnary search on convex shit?
 - 2-sat
 - suffix sum
 - Meet in the middle
 - BigInt
 - precalc solution locally?
 - 2d cumulative sum
 - think of more cases
 - off by one? <= ?
 - bitmasks?
- Range queries:
 - online or offline?
 - ordering the queries?
 - is update lazyable?
 - fenwick is faster than seg (espically MLEs)
 - MOs algo
 - SQRT Deco
 - Treap
 - Sparse table if offline
- Trees:
 - LCA?
 - think about subtree sizes
 - think about ancesstors
 - diameter?
 - HLD?
 - Euler?
 - with segtree?
- Math:
 - Primes?
 - Sieve?
 - give monther and osama the problem
 - nCk or nPk?
 - Inclusion-Exclusion
 - matrix expo?
 - prob?

- guass elem?
- nim or grundy?
- multiples in log time?
- Strings:
- Hashing?
- Trie?
- DP?
- Common Prefix LCP
- Treap

think about converting SCC to DAG (when needed to connect SCCs), can be helpful to do DP or to get 1-SCC from DAG again

for dijkistra, or in general, when having a cost at final node, then fill the dis array with these costs, and think about pushing all values to heap

Multisource Shortest path -> create a new source, s0, and add an edge (with length 0) from s0 to each of your starting vertices. Then, run any shortest-paths algorithm starting from s0 to compute the distance from s0 to each other vertex.