

MUSTAQIL ISHLASH UCHUN MASALALAR YECHIMLARI

1-masala. Yechim (Dijkstra, 2 holat)

```
long[][] dist = new long[n+1][2];
for (int i=1;i<=n;i++) Arrays.fill(dist[i], Long.MAX_VALUE);
dist[1][0] = 0;

PriorityQueue<long[]> pq =
    new PriorityQueue<>(Comparator.comparingLong(a -> a[0]));
pq.add(new long[]{0,1,0});

while (!pq.isEmpty()) {
    long[] cur = pq.poll();
    long d = cur[0]; int u = (int)cur[1], used = (int)cur[2];
    if (d > dist[u][used]) continue;

    for (Edge e: g[u]) {
        if (dist[e.v][used] > d + e.w) {
            dist[e.v][used] = d + e.w;
            pq.add(new long[]{dist[e.v][used], e.v, used});
        }
        if (used == 0) {
            long nd = d + e.w/2;
            if (dist[e.v][1] > nd) {
                dist[e.v][1] = nd;
                pq.add(new long[]{nd, e.v, 1});
            }
        }
    }
}
System.out.println(Math.min(dist[n][0], dist[n][1]));
```

2-masala. Yechim (MST + LCA)

- Kruskal bilan MST
- MST daraxtida LCA + max-edge tayyorlash
- Har MST bo'lmagan qirra (u, v, w) uchun:
sikl = $w + \text{maxEdge}(u, v)$
- Eng kichigini oling

3-masala. Yechim (Kosaraju + DAG DP)

- SCC topish
- SCC qiymatlarini yig'ish
- Siqilgan DAG'da DP:

```
dp[s] = sccValue[s] + max(dp[prev])
```

4-masala. Yechim (DP over subsets)

`dp[mask] = nechta tartib`

- mask dagi tugunlar joylashtirilgan
- `indeg=0` bo'lganlarni qo'shib borish

5-masala. Yechim (Segment Tree)

- Build
- Point update
- Range minimum query
 $O(\log n)$ har so'rov

6-masala. Yechim (BFS + counting)

`ways[v] += ways[u] // faqat eng qisqa qatlamda`

7-masala. Yechim (Dinic)

- Level graph (BFS)
- Blocking flow (DFS)
- $O(\sqrt{V} \cdot E)$ amaliy tez

8-masala. Yechim (Reverse processing + DSU)

- O'chirishlarni teskari tartibda qo'shish
- DSU bilan tekshirish

9-masala. Yechim (Binary Lifting LCA)

`dist(u,v)=depth[u]+depth[v]-2*depth[lca]`

10-masala. Yechim (Bitmask DP)

`dp[u][mask] = minimal cost`

mask — tashrif buyurilgan tugunlar