MUSTAQIL ISHLASH UCHUN MASALALAR YECHIMLARI

1-masala. Yechim (Dijkstra, 2 holat)

```
import heapq

n, m = map(int, input().split())
g = [[] for _ in range(n+1)]
for _ in range(m):
    u, v, w = map(int, input().split())
    g[u].append((v, w))
    g[v].append((u, w))

INF = 10**18
dist = [[INF]*2 for _ in range(n+1)]
dist[1][0] = 0
pq = [(0, 1, 0)]

while pq:
    d, u, used = heapq.heappop(pq)
    if d > dist[u][used]: continue
    for v, w in g[u]:
        if dist[v][used] > d + w:
            dist[v][used] = d + w
            heapq.heappush(pq, (dist[v][used], v, used))
        if used == 0:
            nd = d + w//2
            if dist[v][1] > nd:
                dist[v][1] = nd
                heapq.heappush(pq, (nd, v, 1))

print(min(dist[n]))
```

2-masala. Yechim (MST + max-edge-on-path)

```
# Kruskal MST, keyin har MST qirrasi bo‘yicha max-edge LCA bilan
# Har bir non-MST qirrasi uchun sikl og‘irligi = w +
path_max(u,v)
# Eng kichigini olamiz
```

(Izoh: MST quriladi, daraxtda LCA + max-edge so‘rovlar tayyorlanadi.)

3-masala. Yechim (Kosaraju + DAG DP)

```
# SCC topish → scc_id
# scc_graph qurish → DAG
# dp[s] = scc_value[s] + max(dp[prev])
```

4-masala. Yechim (DP over subsets)

```
# indeg mask bilan
# dp[mask]: nechta tartib
# maskda bo‘lmagan, indeg=0 bo‘lgan tugunlarni qo‘shib borish
```

5-masala. Yechim (Segment Tree)

```
# build, update, query min — O(log n)
```

6-masala. Yechim (BFS + counting)

```
from collections import deque
MOD = 10**9+7

n, m = map(int, input().split())
g = [[] for _ in range(n+1)]
for _ in range(m):
    u, v = map(int, input().split())
    g[u].append(v); g[v].append(u)

dist = [-1]*(n+1)
ways = [0]*(n+1)
dq = deque([1])
dist[1] = 0
ways[1] = 1

while dq:
    u = dq.popleft()
    for v in g[u]:
        if dist[v] == -1:
            dist[v] = dist[u] + 1
            ways[v] = ways[u]
            dq.append(v)
        elif dist[v] == dist[u] + 1:
            ways[v] = (ways[v] + ways[u]) % MOD

print(ways[n])
```

7-masala. Yechim (Dinic)

```
# Dinic algoritmi: level graph + blocking flow
# n ≤ 200 uchun yetarli
```

8-masala. Yechim (DSU rollback / reverse processing)

```
# O‘chirishlarni teskari tartibda qo‘shib borish
# DSU bilan javoblarni yig‘ish
```

9-masala. Yechim (Binary Lifting LCA)

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

10-masala. Yechim (Bitmask DP — TSP)

```
INF = 10**18
n = int(input())
w = [list(map(int, input().split())) for _ in range(n)]
```

```
dp = [[INF]*(1<<n) for _ in range(n)]
dp[0][1] = 0

for mask in range(1<<n):
    for u in range(n):
        if dp[u][mask] < INF:
            for v in range(n):
                if not (mask>>v)&1:
                    dp[v][mask|(1<<v)] = min(dp[v][mask|(1<<v)],
                                             dp[u][mask] +
w[u][v])

full = (1<<n)-1
ans = min(dp[u][full] + w[u][0] for u in range(n))
print(ans)
```