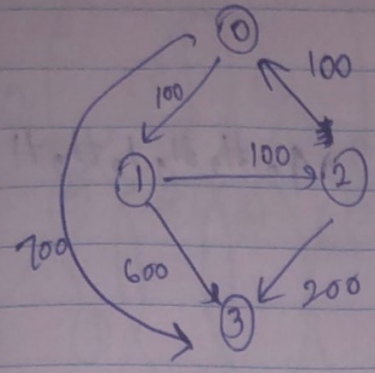
* Cheapest Flights Within K Stops



K=2 from 1 to 3.

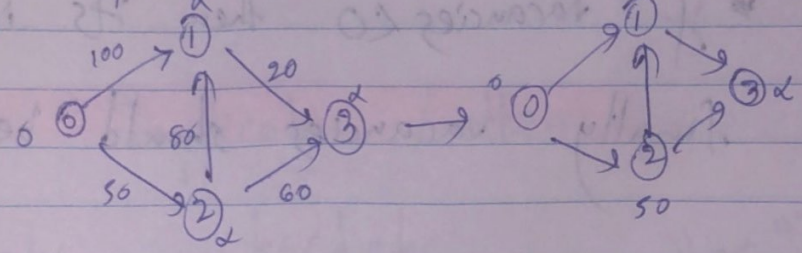
weighted graph representation

1) Adjacency matrix
2) Adjacency List as {node, weight}

- BFS
  - O(V + E(K))
  - O(N + E(K))

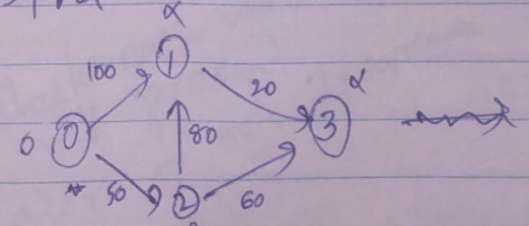
if price + distance < current price

- Bellman Ford
  - O((N+E).K)
  - O(N)



needs 2 lists of distance to keep current updatings & previous.

- Dijkstra
  - O(N + E K log(E K))
  - O(N + E V)



explore adjacents & take the one with min distance. (priority Queue)