Day 18 Dijkstra

ITSRUNTYM

What is Dijkstra's Algorithm?

Dijkstra's algorithm is used to find the **shortest path from a source node to all other nodes** in a **weighted, non-negative graph**.

Key Concepts:

- Works only with **non-negative weights**
- Uses **greedy approach**
- Uses **Priority Queue (Min-Heap)** for optimization

Use Case Example:

We'll use this undirected, weighted graph:

Step-by-Step Working of Dijkstra

1. Set dist[source]=0 and all other distances as infinity.
2. Push the source node into the min heap as a pair <distance, node> → i.e., <0, source>.
3. Pop the top element (node with the smallest distance) from the min heap.
4. For each adjacent neighbor of the current node:
5. Calculate the distance using the formula:
   dist[v] = dist[u] + weight[u][v]
   If this new distance is shorter than the current dist[v], update it.
   Push the updated pair <dist[v], v> into the min heap
6. Repeat step 3 until the min heap is empty.
7. Return the distance array, which holds the shortest distance from the source to all nodes.

#	Problem Name	LeetCode Link	Notes
1	**Network Delay Time**	🔗 743. Network Delay Time	Classic Dijkstra on directed weighted graph

2	**Path With Minimum Effort**	🔗 1631. Path With Minimum Effort	Dijkstra using min-heap with custom edge cost (effort)
3	**Cheapest Flights Within K Stops**	🔗 787. Cheapest Flights Within K Stops	Dijkstra with constraint on stops (can use BFS + PQ)
4	**The Maze II**	🔗 505. The Maze II	Dijkstra-like traversal in grid (ball rolls until wall)
5	**Minimum Cost to Reach Destination in Time**	🔗 1928. Minimum Cost to Reach Destination in Time	Dijkstra with both cost and time constraints
6	**Find the City With the Smallest Number of Neighbors at a Threshold Distance**	🔗 1334. Find the City With the Smallest Number of Neighbors	Run Dijkstra from every node
7	**Minimum Number of Work Sessions to Finish the Tasks**	🔗 1986. Minimum Number of Work Sessions	Can be solved via DP + Dijkstra in some approaches
8	**Kth Smallest Prime Fraction**	🔗 786. K-th Smallest Prime Fraction	Uses min-heap with pairs (a/b) – Dijkstra idea
9	**Reach the Safest Path in a Grid**	🔗 2812. Find the Safest Path in a Grid	Dijkstra from all threats first, then main traversal
10	**Minimum Time to Reach Destination Without Drowning**	🔗 1976. Number of Ways to Arrive at Destination	Dijkstra + path counting