Finding the shortest paths

→ Given an (edge-) weighted graph & two vertices in it.

→ Find the cheapest (minimum possible weight) path between them,
or

→ Report that one does exist.

& a vertex s in it

find the cheapest (minimum possible weight) paths from s to all other vertices.

Greedy Algorithm

Dijkstra's algorithm

↓

by BFS with a twist

- The queue is replaced with a minimum priority queue.
- with an additional operation decrease-priority (vertex, new-priority)

priority(v) = distance(start, v) via finished vertices only.

priority(v) = ∞ if no such path.

! 路径权重是累加的.

dist[v] = min(dist[v], dist[u] + weight(u,v))

⟶ grows path by one edge at a time.

IDEA: Every time we extract-min, we get the next vertex closest to the start.