Shortest paths in weighted graphs

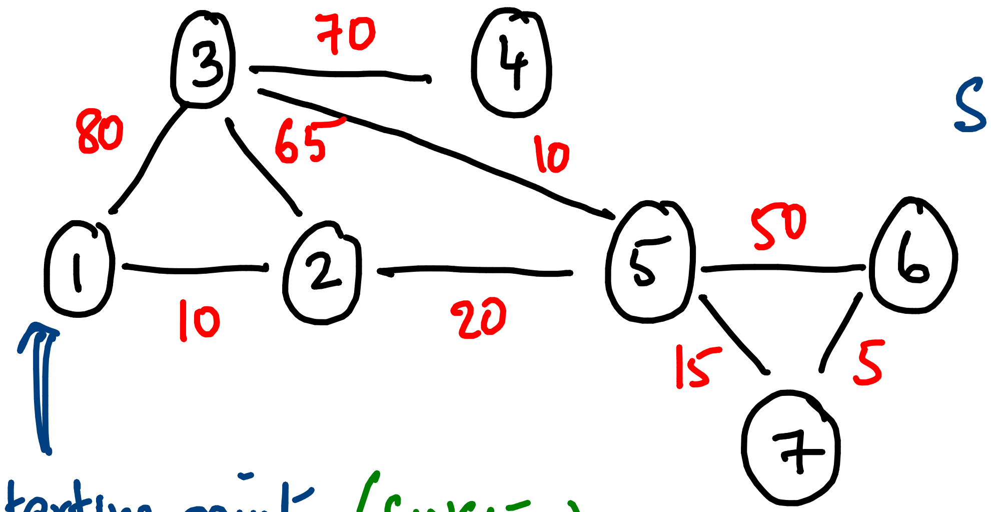
$G = (V, E)$, $w : E \to \mathbb{R}$ — weight/cost of each edge

Distance/Price/Time

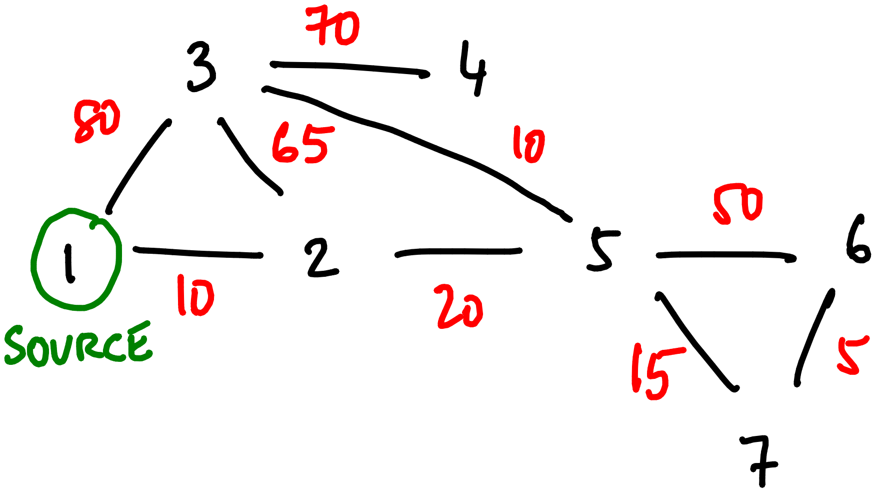
Assume non negative weights

ALL PAIRS shortest paths

SINGLE SOURCE shortest paths



starting point (SINGLE SOURCE)



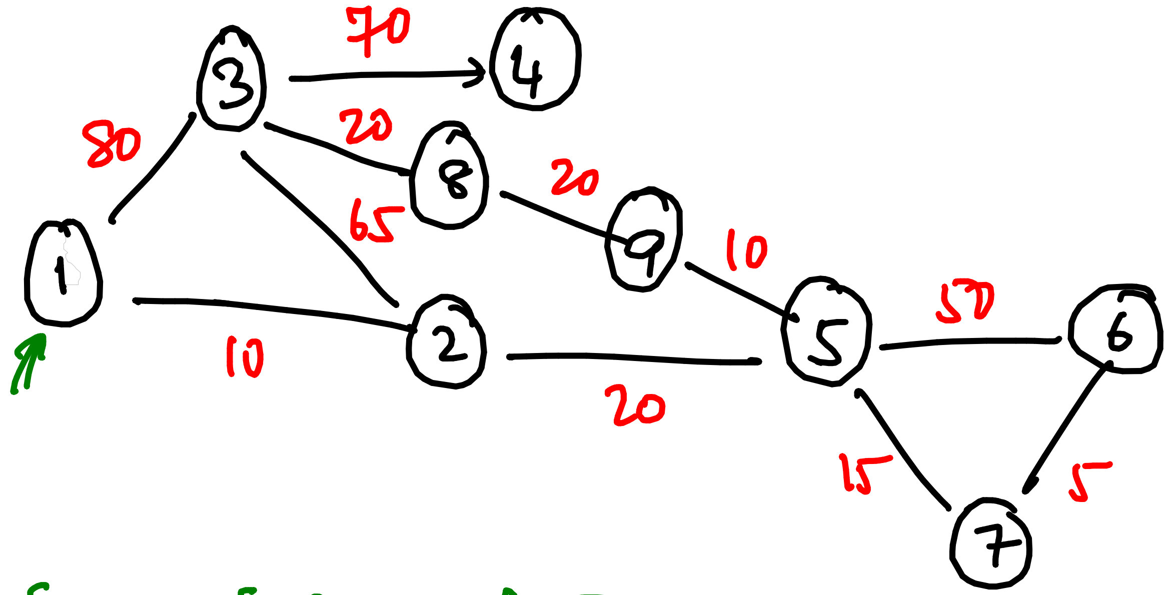
Goal: Compute Distance(1, j) for every j

Suggested strategy

1 [0] → 2 [10] → 5 [30] → 3 [40] → 4 [110]

↓

7 [45] → 6 [50]

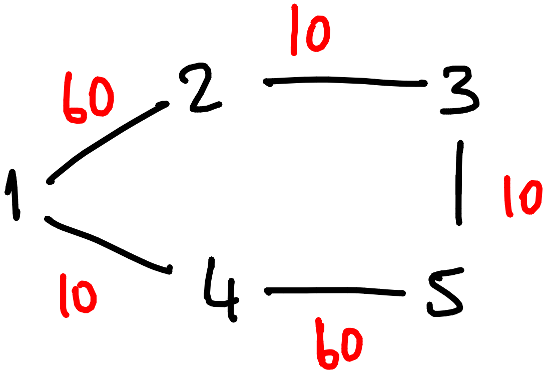


1[0] → 2[10] → 5[30] → 9[40] → 8[60]

3[75]

└ 4[115]

↓ 7[45] – 6[50]

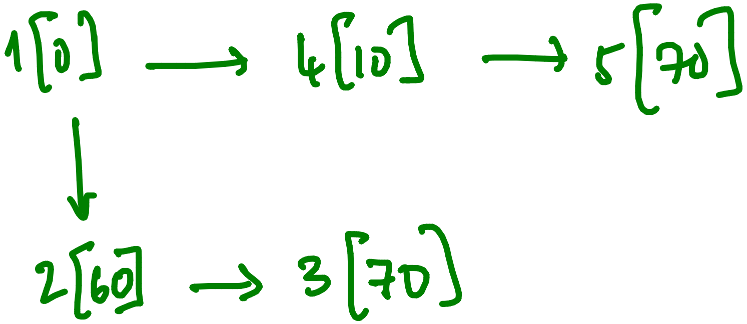


What are we keeping track of?

INVARIANT

At a given point

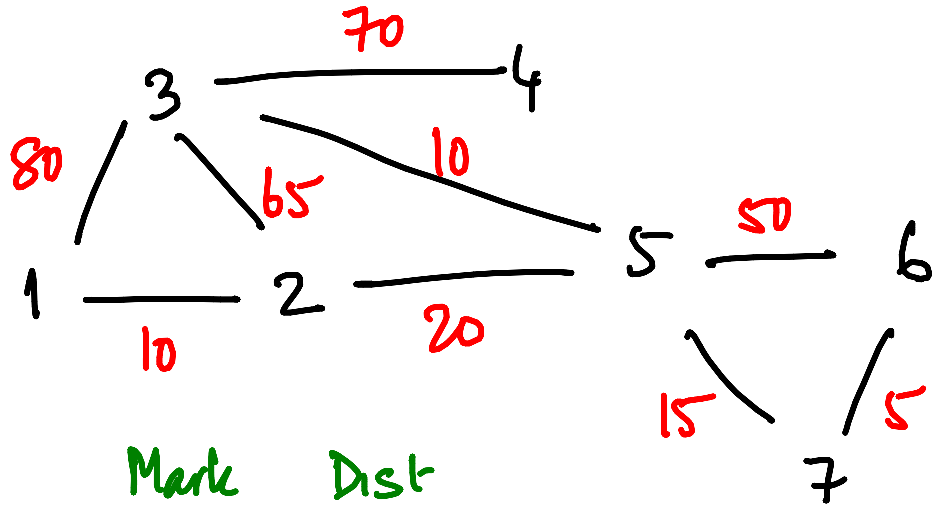
- Marked some vertices
- Assigned distances to marked vertices
- Assigned estimates to nbrs of marked vertices



For each vertex, maintain

```
    Marked(i)    – initially 0
    Distance(i)  – initially ∞

Assume source is 1
    Assign Distance(1) = 0
    Repeat n times (assume G connected)
        Choose i s.t. Marked[i]==0, Distance[i] is min
        Set Marked[i]=1,
        For each (i,j), update Distance[j] = min(Distance[j],
          s.t. Marked[j]==0                       Dist[i] + W(i,j))
```



	Mark	Dist
1	~~0~~ 1	~~∞~~ 0
2	~~0~~ 1	~~∞~~ 10
3	~~0~~ 1	~~∞~~ ~~80~~ ~~75~~ 40
4	~~0~~ 1	~~∞~~ 110
5	~~0~~ 1	~~∞~~ 30
6	~~0~~ 1	~~∞~~ ~~80~~ 50
7	~~0~~ 1	~~∞~~ 45

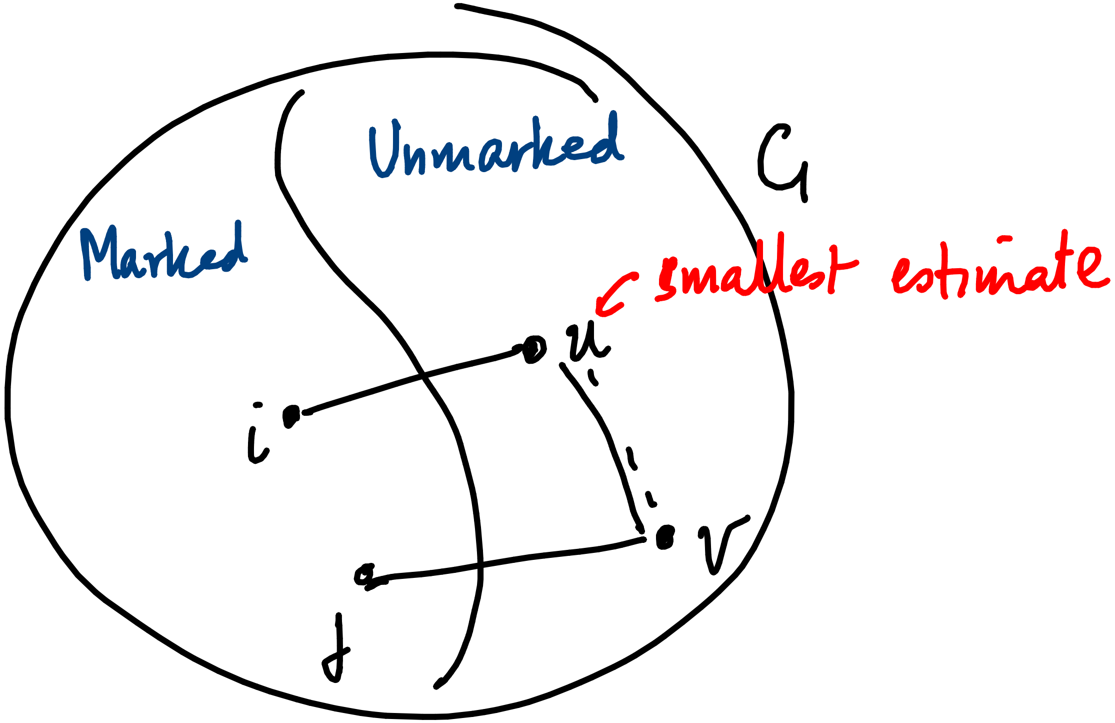
DIJKSTRA'S

SINGLE SOURCE

SHORTEST PATH

ALGORITHM

Correctness



estimate(v) $\geq$ estimate(u) currently

$\therefore$ distance(v) + w(v,u) $\geq$ distance(u) – crucial that wts are not negative

Efficiency

Initialize Marked[], Distance[] $O(n)$

repeat n times

- Find best estimate to mark
- Update estimates of newly marked vertex

Adj Matrix, both $O(n)$] Overall $O(n^2)$

Still $O(n)$??

↓

need a better data structure

Adj List $O(m)$

Priority Queue

Each element has a priority

- Typically set when entering queue

Remove highest priority item

Insert new elements

Dijkstra's algo – priority = estimate

Also have to update priorities

DIJKSTRA = BFS with PRIORITY QUEUE