P_3: Design Experience

Dijkstra's Algorithm: Greedy Approach

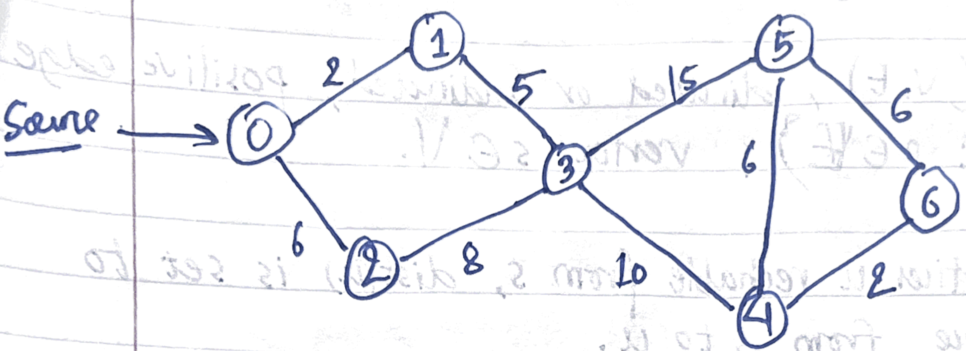
Input: Graph $G = (V, E)$, directed or undirected; positive edge lengths $\{l_e : e \in E\}$; vertex $s \in V$.

Output: For all vertices u rechable from s, dist(u) is set to the distance from s to u.

Pseudocode

```
Procedure dijkstra(G, l, s):
    for all u ∈ V:
        dist(u) = ∞
        prev(u) = nil
    dist(s) = 0
    H = makequeue(V)        [∵ Using dist-val as keys]
    While H is not empty:
        u = deletemin(H)
        for all edges (u,v) ∈ E:
            if dist(v) > dist(u) + l(u,v):
                dist(v) = dist(u) + l(u,v)
                prev(v) = u
                decreasekey(H,v)
```

① Hand-solved example :



Problem is to find the shortest path from source (i.e. 0) to all other nodes.

Initial

0 : 0
1 : ∞
2 : ∞
3 : ∞
4 : ∞
5 : ∞
6 : ∞

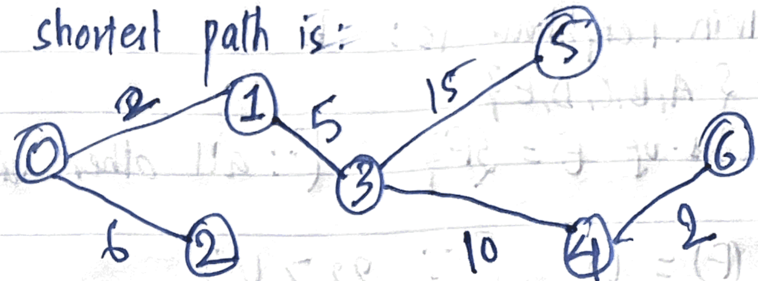
unvisited nodes : {0, 1, 2, 3, 4, 5, 6}

→ The algorithm is only complete when all the nodes are visited and added to path

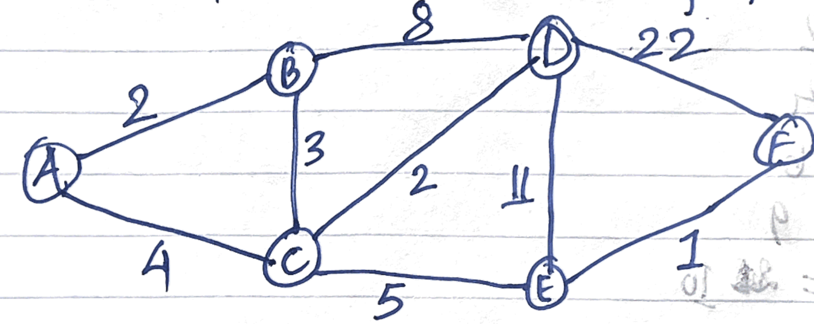
0 : 0
1 : $\cancel{\infty}$ 2
2 : $\cancel{\infty}$ 6
3 : $\cancel{\infty}$ 7
4 : $\cancel{\infty}$ 17
5 : $\cancel{\infty}$ 22
6 : $\cancel{\infty}$ 19

unvisited = {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~}

Order of visiting nodes → 0 → 1 → 2 → 3 → 4 → 6 → 5 so, the shortest path is:



② <u>Another Example</u> (detailed discussion of pseudocode)



- Initial distance assumption is ∞ for all nodes, except A [$\because$ 0 for A]
- Adj nodes of A = {B, C} Visited = [A]

- cost of each neighbour ⇒ cost (B) = Cost (A) + dist (A → B)
  = 0 + 2 = 2

Similarly,
  Cost (C) = 0 + 4 = 4

- Finding the min. cost neighbour out of {2, 4} is 2 : i.e. take B nod
  Similar approach like above,
  adj nodes of B : {C, D} [$\because$ A has been visited already]

  Cost (C) = 4 [$\because$ 2+3 = 5 > 4]
  Cost (D) = 10 [$\because$ 0+2+8]

→ Next is C ⇒ adj. nodes = {D, E}
  cost (D) = 4+2 = 6
  Cost (E) = 4+5 = 9

Next is D. Adj. nodes of D = {E, F}
  cost (E) = 9
  cost (F) = 28

Visited = {A, B, C}

The next min. neighbour is : E

Visited = { A, B, C, D, E }

so, Adj nodes of E = {F} [∵ all other visited already]

Cost (F) = 10 [∵ 22 > 10]

Thus,

Cost (A) = 0 [Source]

cost (B) = 2

Cost (C) = 4

cost (D) = 6

Cost (E) = 9

Cost (F) = ~~22~~ 10

The shortest Path is :