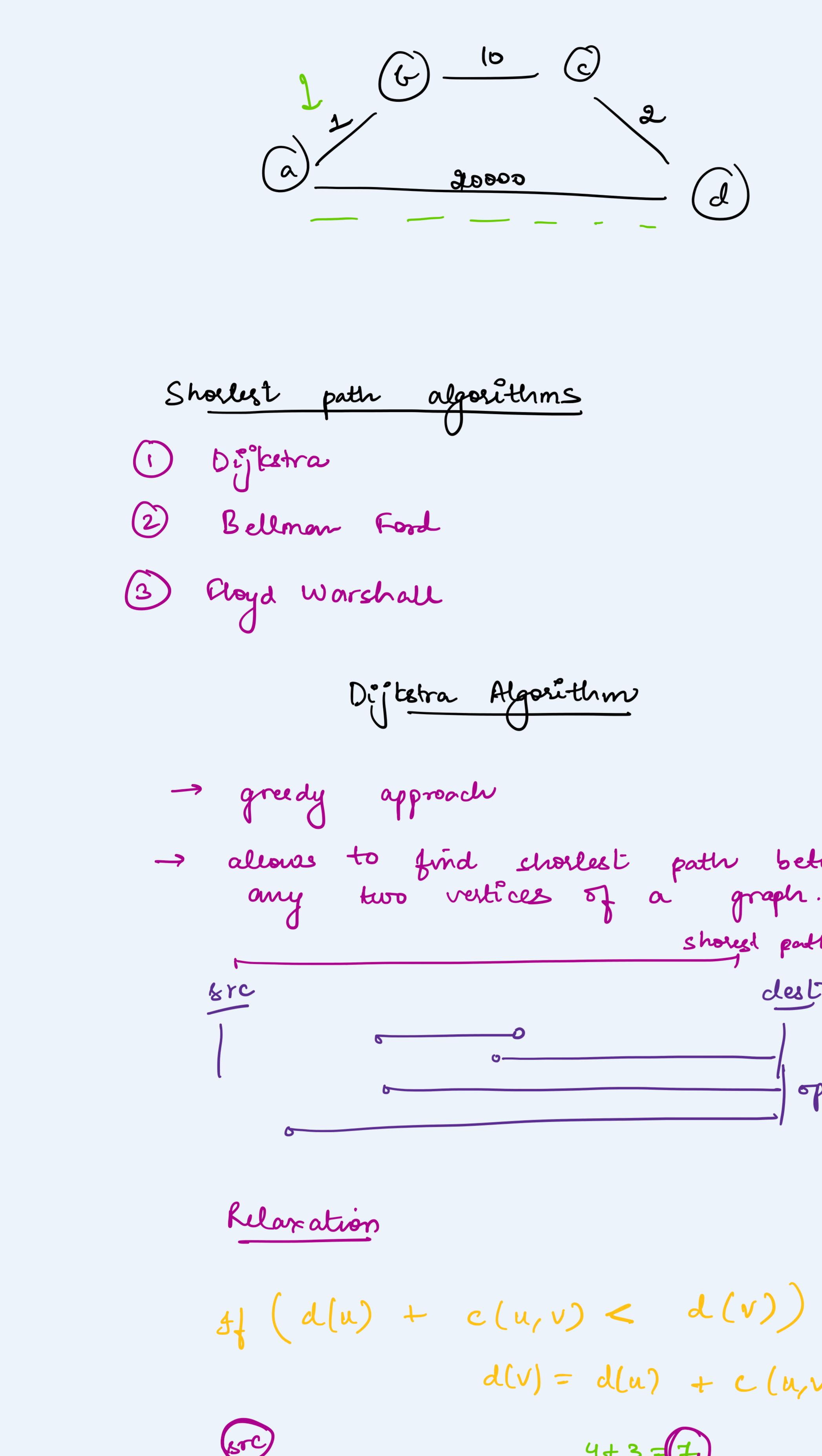
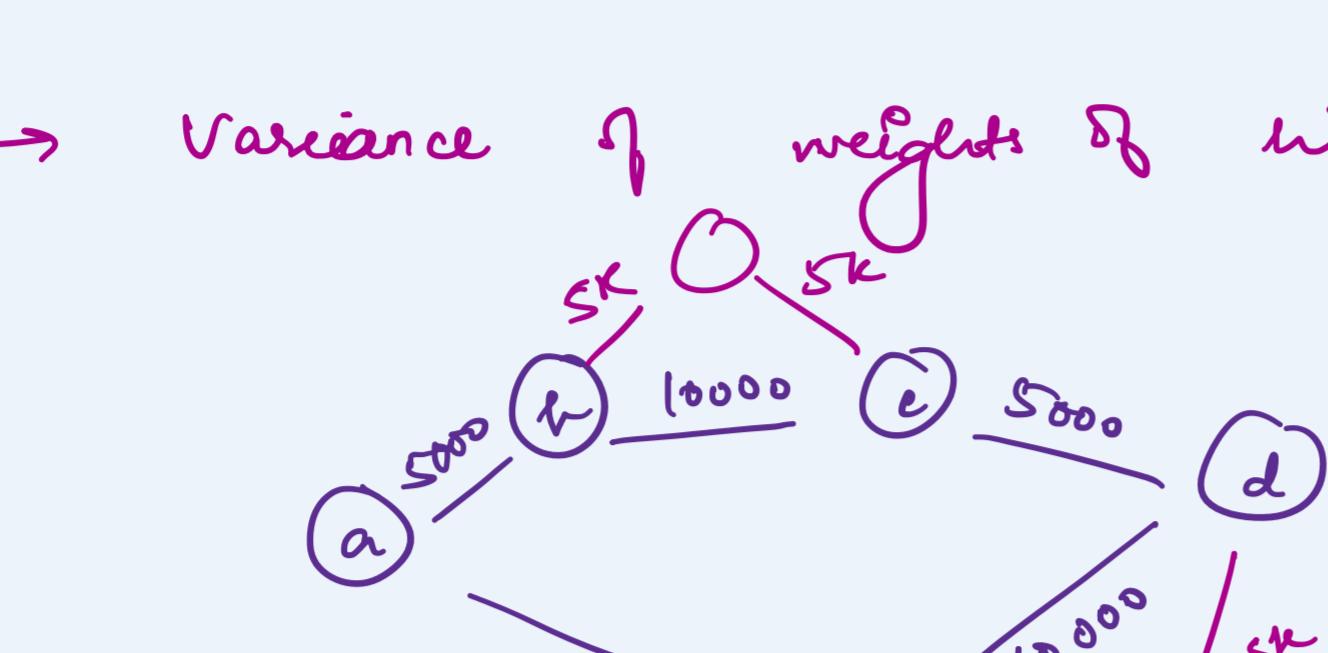
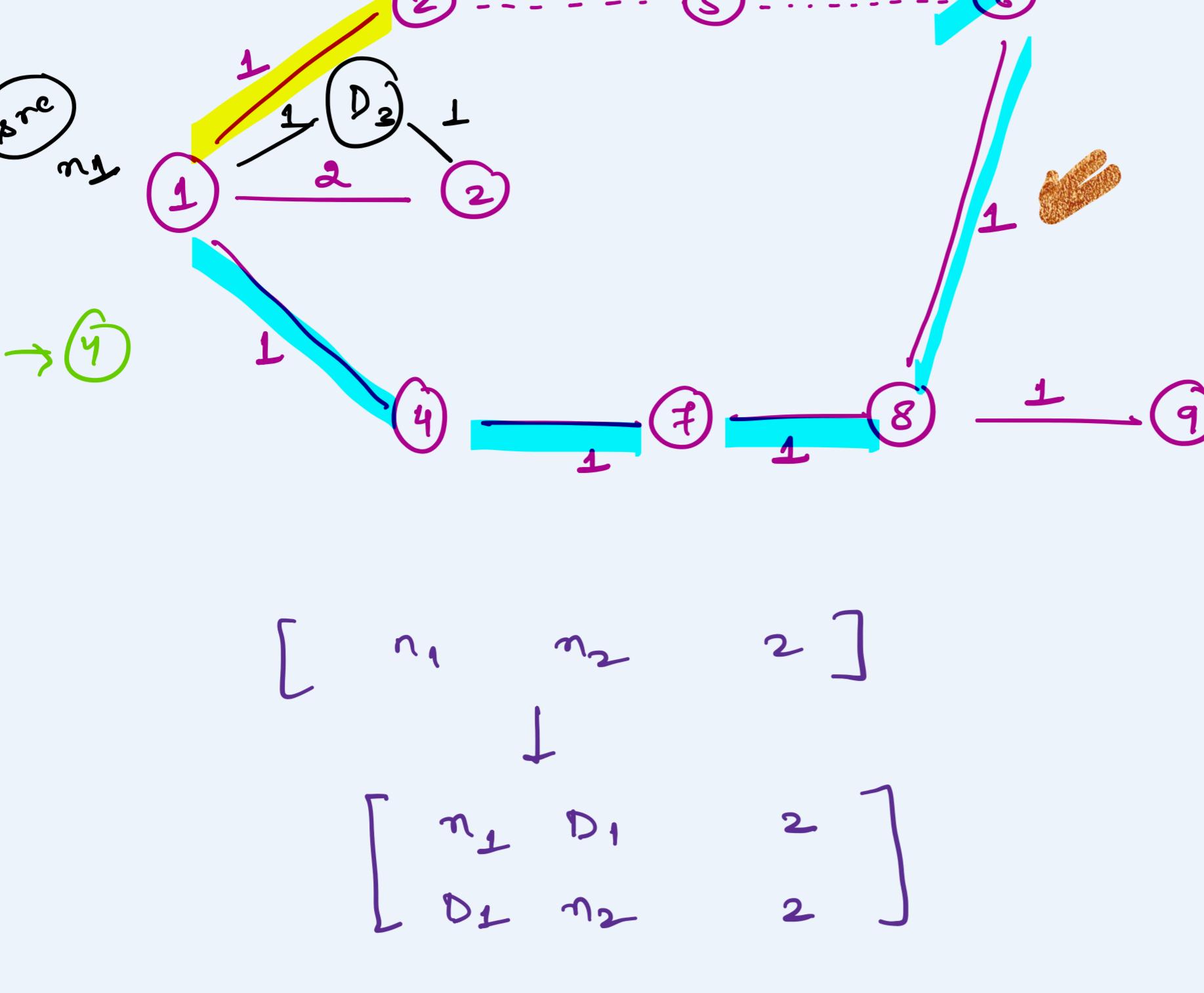


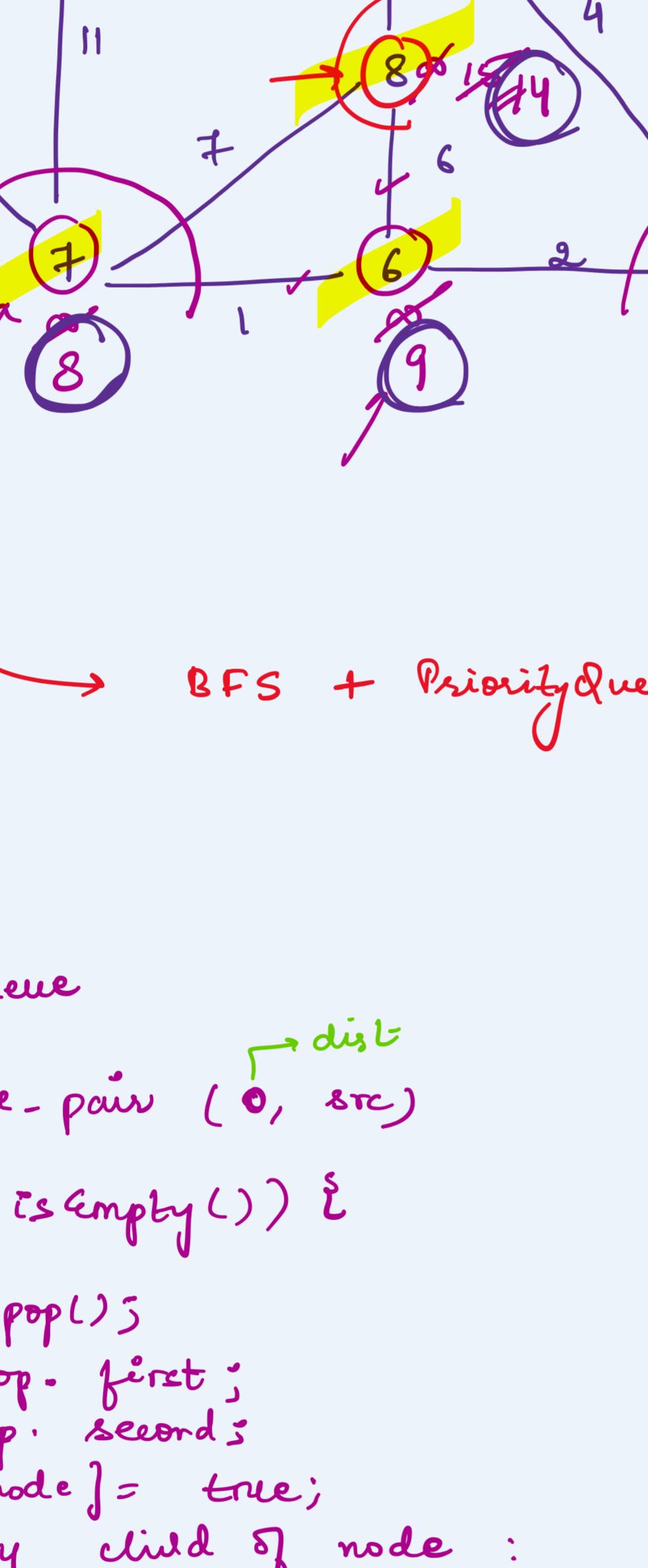
→ Breadth first search  
(level by level)

BFS can be used to find the shortest path between any two nodes.



↑ nodes      ↑ edges      ⇒ ↑ space

→ Variance of weights is high



### Shortest path algorithms

① Dijkstra

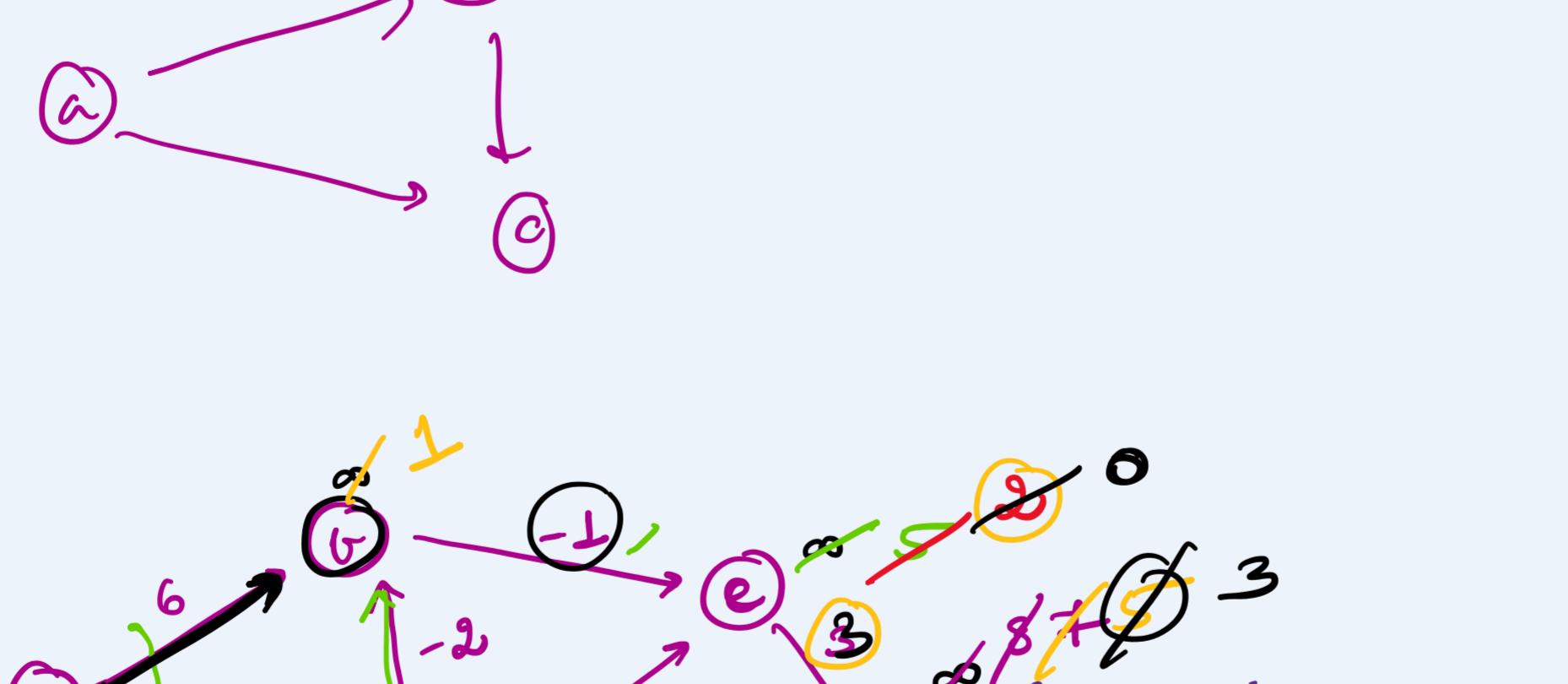
② Bellman Ford

③ Floyd Warshall

### Dijkstra Algorithm

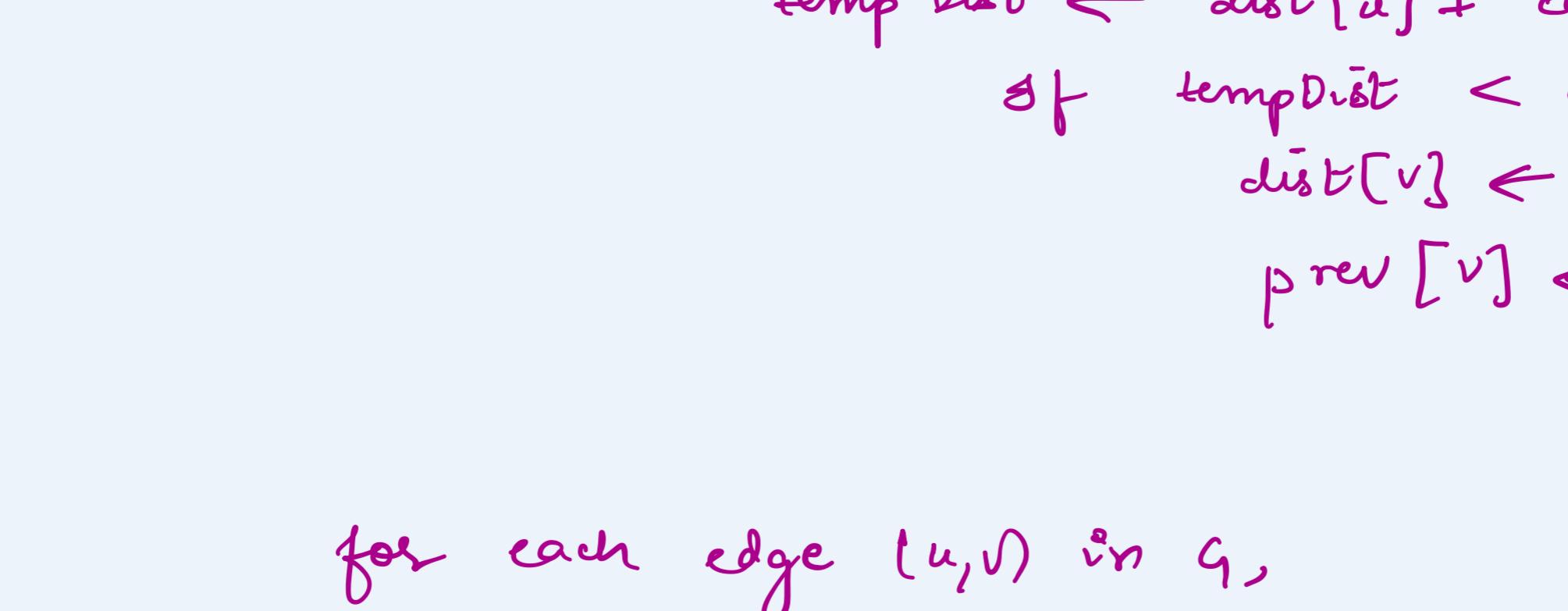
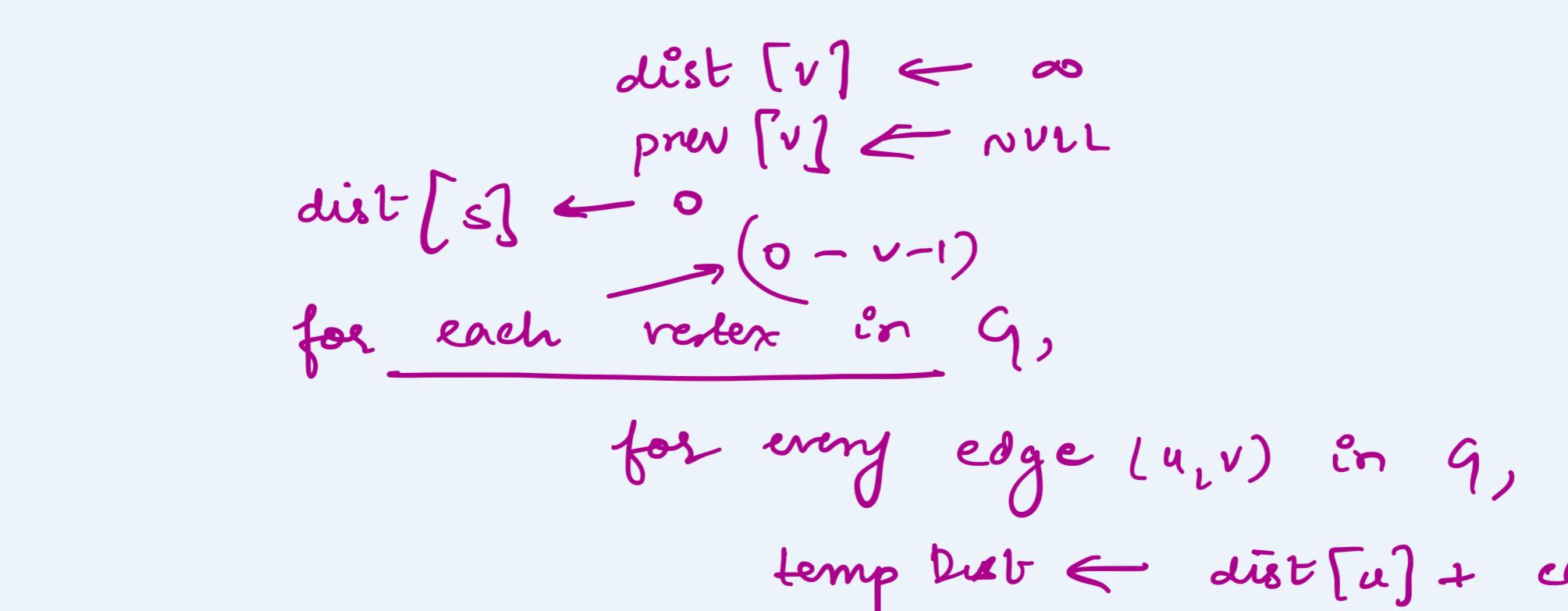
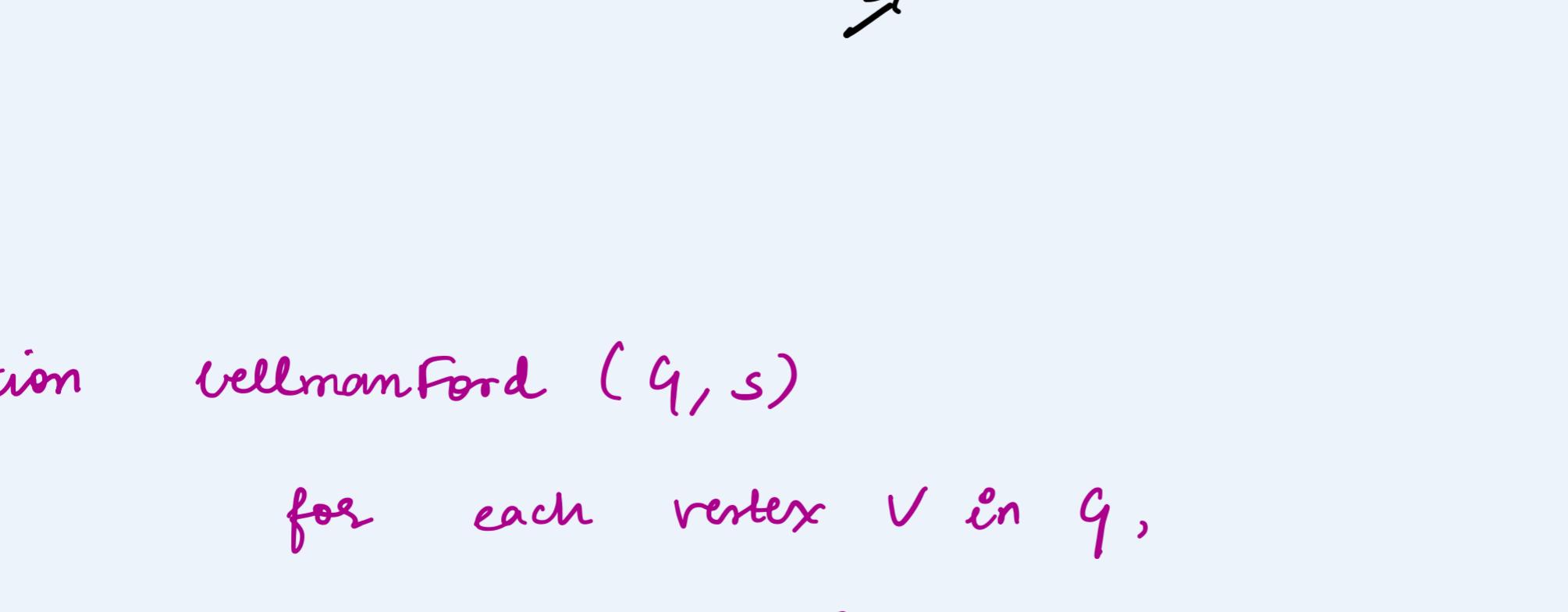
→ greedy approach

→ allows to find shortest path between any two vertices of a graph.



### Relaxation

$$\text{if } (d(u) + c(u,v) < d(v)) \\ d(v) = d(u) + c(u,v)$$



function Dijkstra (G, s)

for each vertex v in G,

dist[v] ← ∞

prev[v] ← null

dist[s] ← 0

for each vertex v in G,

for each edge (u, v) in G,

tempDist ← dist[u] + edgeWeight(u, v)

if tempDist < dist[v]

dist[v] ← tempDist

prev[v] ← u;

for each edge (u, v) in G,

if (dist[u] + edgeWeight(u, v) < dist[v])

"negative cycle exists"

→ Shortest Bridge



→ More stones removed.

→ Shortest Path



→ Shortest Path

