Where to use the DIJKSTRA ALGORI. with PROOF

LinkedIn @manojofficialmj

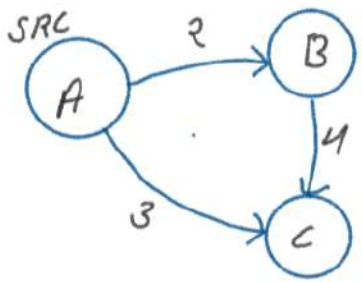
📂 1. Dijkstra Algorithm

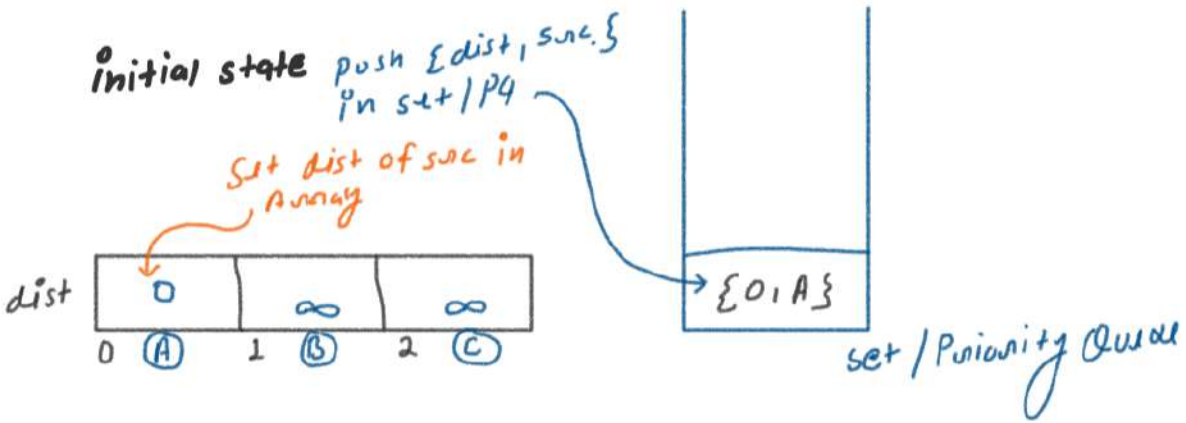
🚀 *Dijkstra Algorithm: Source, Destination, Minimum*

Note 1: *It is used to find all shortest path from source to all nodes in a negative and positive weighted, directed or undirected graph.*

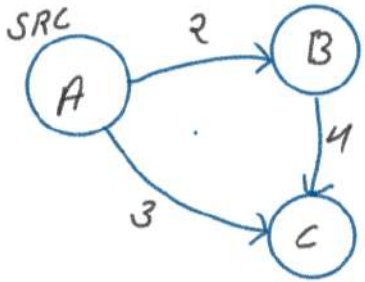
Note 2: *It is not used to when a graph have the negative cycle because of occurring the infinite loop*

Example 1: *positive weighted directed graph*





DRY RUN

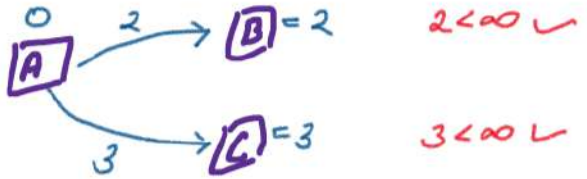


Iteration 1

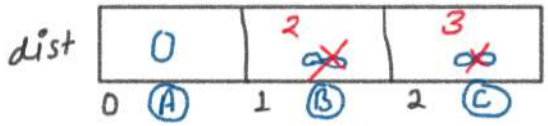
STEP1 Get shortest dist node from set/PQ and pop it

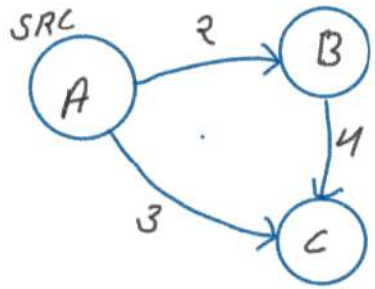
{0, A}

STEP2 update dist Array and create new entry in set also



set / Priority Queue



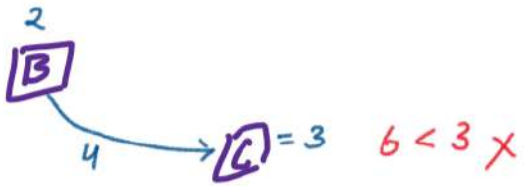


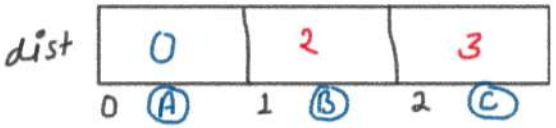
Itration 2

STEP1 Get shortest dist node from set/PQ and pop it

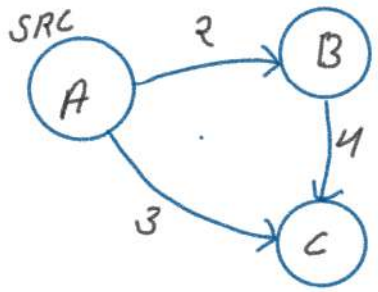
{2, B}

STEP2 update dist Array and create new Entry in set when





set / Priority Queue



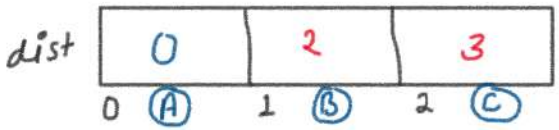
Iteration3

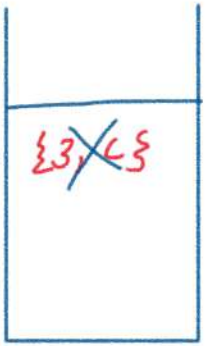
STEP1 Get shortest dist node from set/PQ and pop it

{3, C}

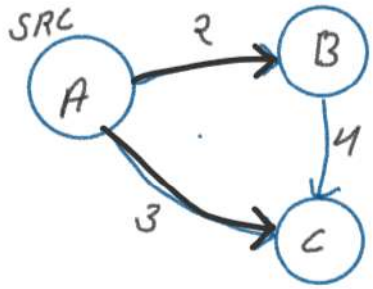
STEP2 update dist Array and create new entry in set also

3
C → Yanha se kahi ja Hi nahi skte hai





set / Priority Queue

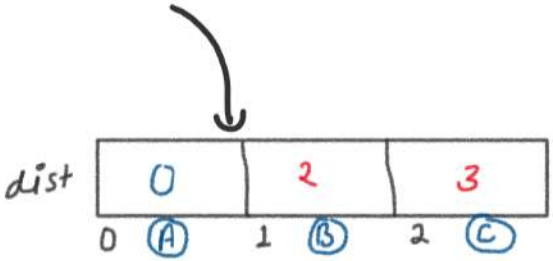


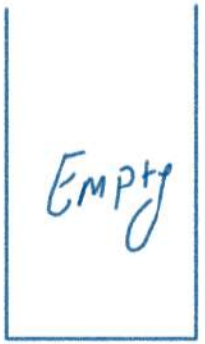
Iterations

STEP 1 Get shortest dist node from set/PQ and pop it

↳ Ab set/PQ Empty ho chuka hai, so stop the loop

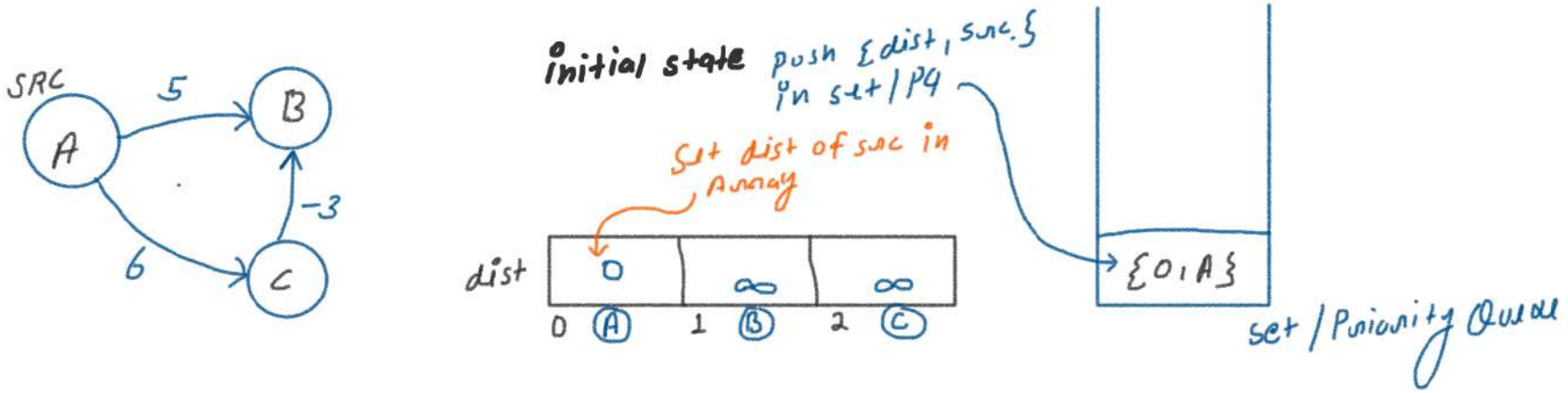
All shortest path from SRC



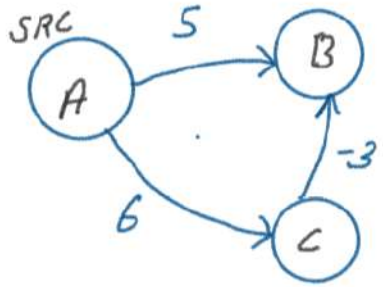


set / Priority Queue

Example 2: *negative weighted directed graph*



DRY RUN

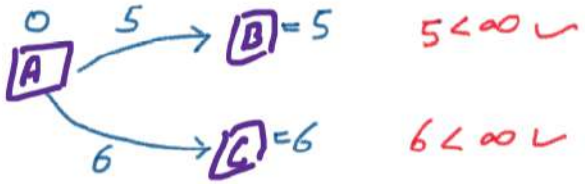


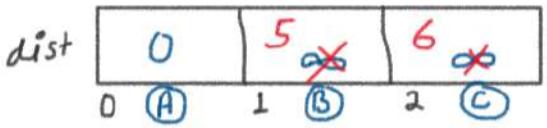
Iteration 1

STEP1 Get shortest dist node from set/PQ and pop it

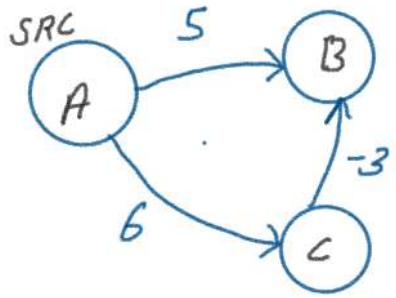
{0, A}

STEP2 Update dist Array and create new entry in set also





set / Priority Queue



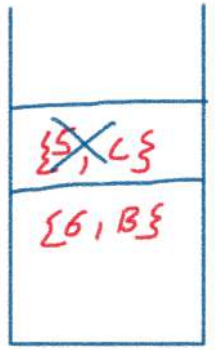
Iteration 2

STEP 1 Get shortest dist node from set/PQ and pop it

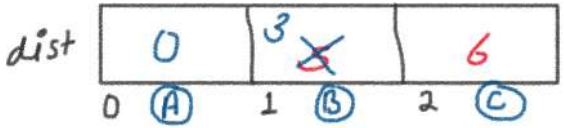
{5, C}

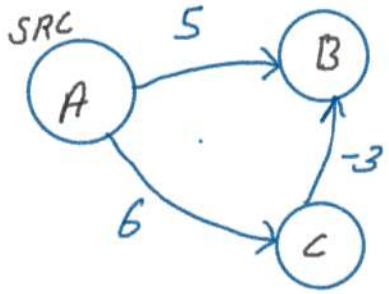
STEP 2 Update dist Array and create new entry in set also

3 < 5 ✓



set / Priority Queue





Iteration3

STEP1 Get shortest dist node from set/PQ and pop it

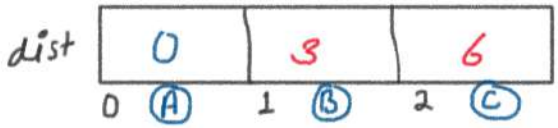
{6, B}

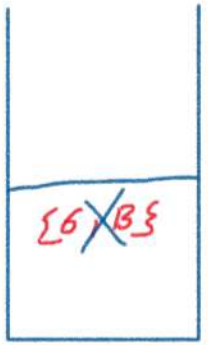
STEP2 update dist Array and create new Entry in set also

3

B

Yanha se kahi ja Hi nahi skte Hai

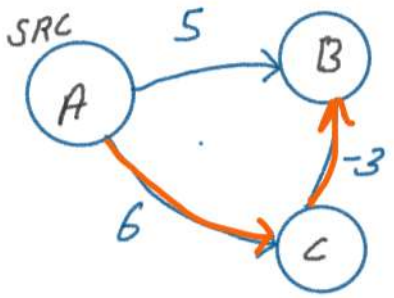




set / Priority Queue

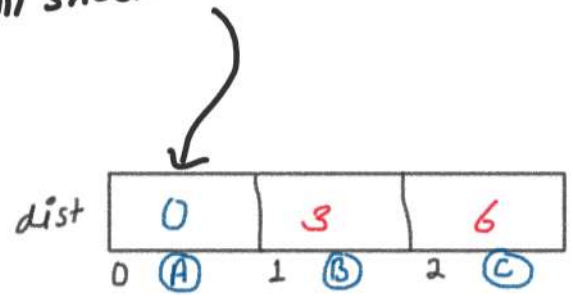
STEP 1 Get shortest dist node from set/PQ and pop it

↳ Ab set/PQ empty ho chuka hai, so stop the loop



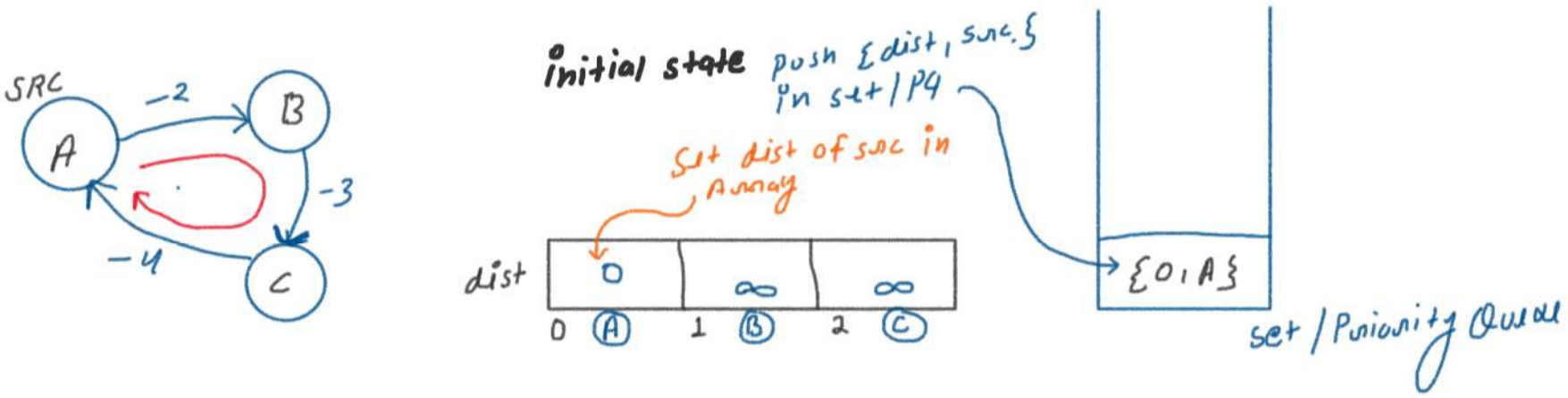
Iteration 4

All shortest path from SRC

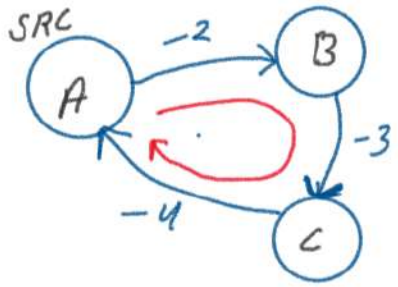


set / Priority Queue

Example 3: negative cycle in directed graph



DRY RUN

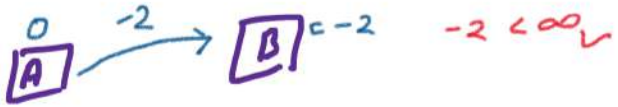


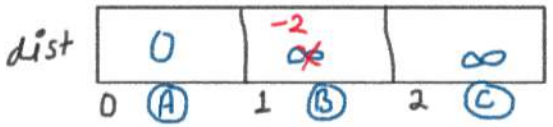
Iteration 1

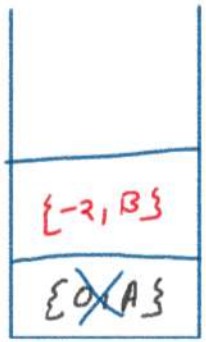
STEP1 Get shortest dist node from set/PQ and pop it

{0, A}

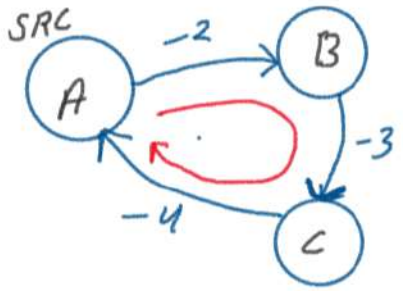
STEP2 update dist Array and create new entry in set also







set / Priority Queue

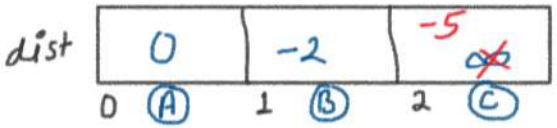


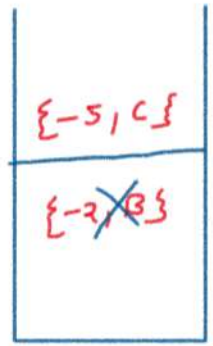
Iteration2

STEP1 Get shortest dist node from set/PQ and pop it

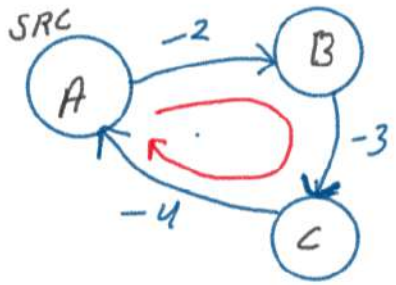
{-2, B}

STEP2 update dist Array and create new entry in set also





set / Priority Queue

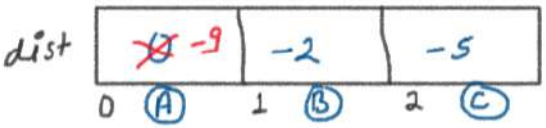


STEP1 Get shortest dist node from set/PQ and pop it

{-5, C}

STEP2 update dist Array and create new entry in set also

Iteration3



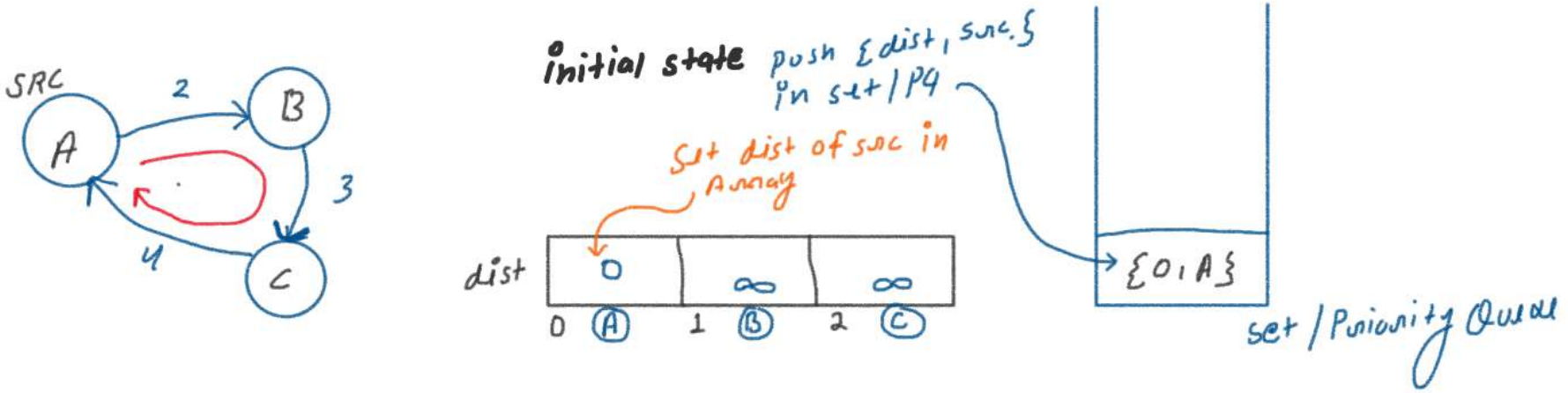
set / Priority Queue

Infinite Loop me Fas jayega

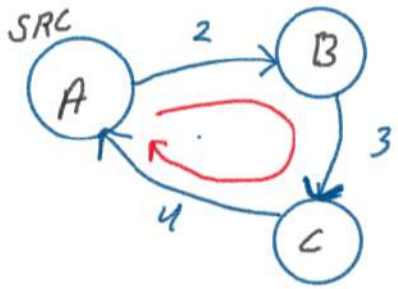
Loop

Jab (-ve) + (-ve) to Hamesh smallest distance mila Rahega as compare to jo dist Array me pahle se Rakha hua hai to Iss Prakar set/PQ EMPTY Nahi Ho Payega

Example 4: *positive cycle in directed graph*



DRY RUN

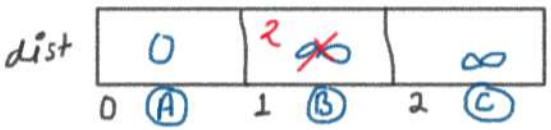


Iteration 1

STEP1 Get shortest dist node from set/PQ and pop it

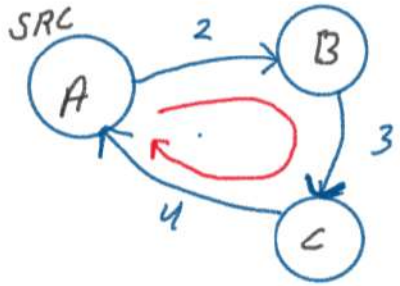
{0, A}

STEP2 update dist Array and create new entry in set also



set / Priority Queue

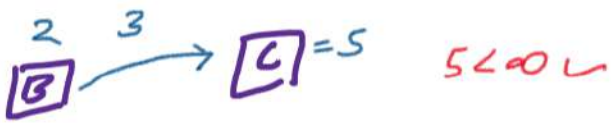
DRY RUN



STEP1 Get shortest dist node from set/PQ and pop it

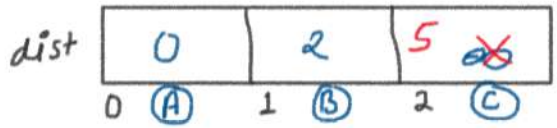
{2, B}

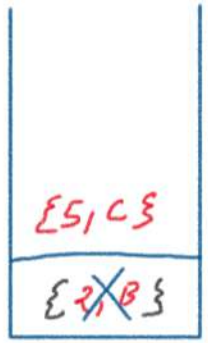
STEP2 update dist Array and create new entry in set also



5 < ∞

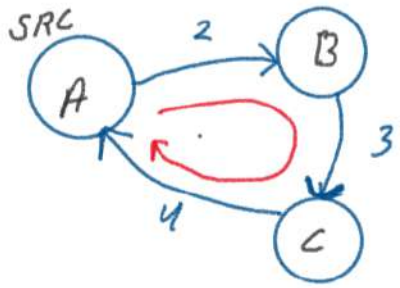
Iteration2





set / Priority Queue

DRY RUN



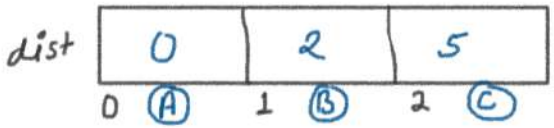
STEP1 Get shortest dist node from set/PQ and pop it

{5, C}

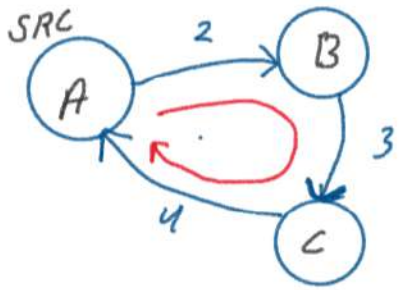
STEP2 update dist Array and create new entry in set when

9 < 0 ✗

Iteration3



set / Priority Queue

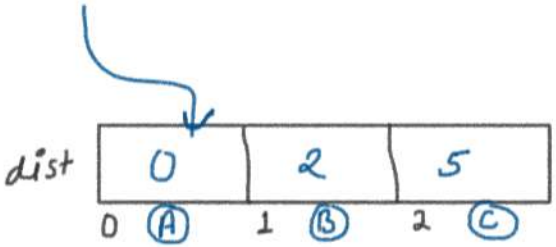


Iterations

STEP 1 Get shortest dist node from set/pq and pop it

↳ Now set/pq is Empty so stop the loop.

All shortest path from src



set / Priority Queue