ANALYSIS OF SHORTEST PATH ALGORITHM

Ajay Sagar Parwani

Dijkstra Algorithm

- Dijkstra Algorithm is algorithm used to find the shortest path from Source vertex to all vertices in a weighted graph with positive integer.

- Weighted & Un weighted Graph

Applications

- Google Maps
- Robots Navigation
- Urban Traffic Planning
- Sub routine in Advance Algorithms
- Routing of Telecommunication messages
- Network Routing Protocols

What we need ?

- There are classes functions and data structures we will need for algorithm
  - Vertex.java
  - MinimumPriorityQueue.java
  - Graph.java
  - ShortestPath.java

Pseudo code

Dijkstra's Pseudo Code

```
DIJKSTRA(G, w, s)
1  INITIALIZE-SINGLE-SOURCE(G, s)
2  S = ∅
3  Q = G.V
4  while Q ≠ ∅
5       u = EXTRACT-MIN(Q)
6       S = S ∪ {u}
7       for each vertex v ∈ G.Adj[u]
8            RELAX(u, v, w)
```

Priority Queue

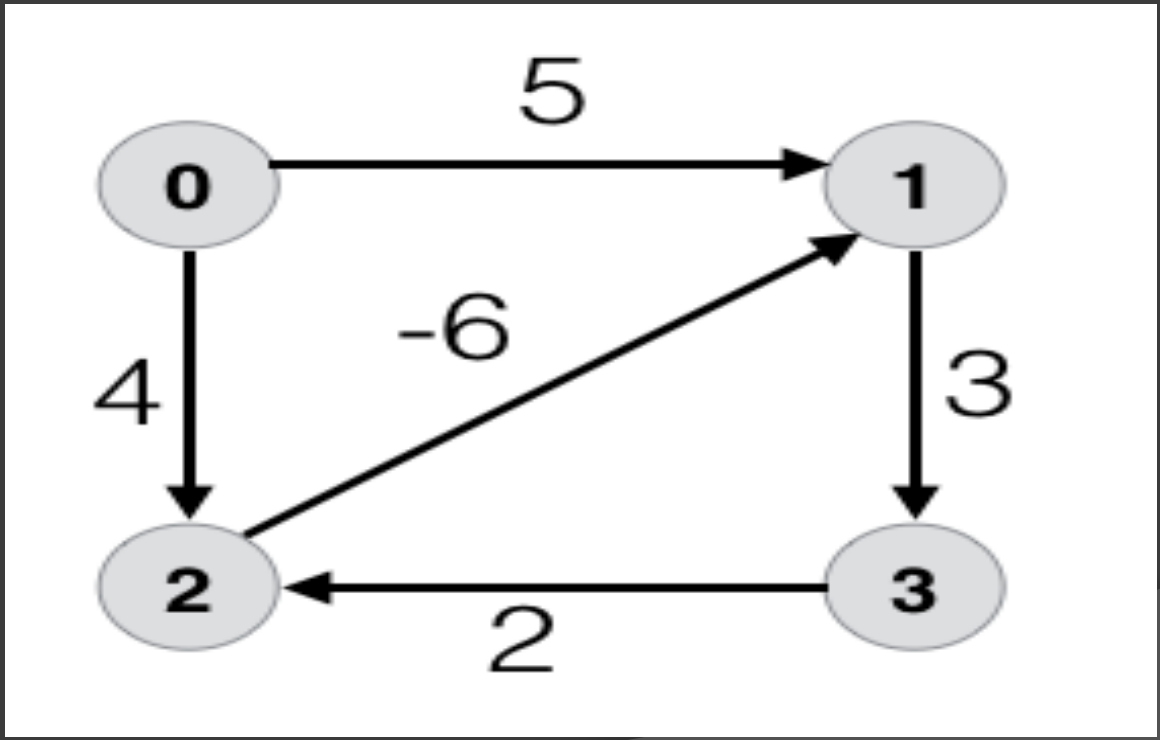
- Minimum Priority Queue
- Structure similar to BST
  - Insertion = O(log n)
  - Deletion = O(log n)
- Fibonacci Heap : Complexity decreases
  - Insertion = O(1)
  - Deletion = O(log n)

Complexity

- Best-Case Scenario: Occurs when each Node contains single neighbor. (V Log V)
- Worst-Case Scenario: Occurs when each node has a route to all vertices.
- In worst case loop repeats V time for vertices and for vertex we are looping through all its edges. (V*E log V)

Why not negative??

- Dijkstra can’t handle to compute negative cycle



Bellman Ford

- Designed by Alfonso Shimbel in 1955
- Negative Weight Edges
- Solution Exists for all ?

Costing

- Lets Do it ?

Questions ?

Thanks