All Source Shortest Path

Learning Objectives

1. Implement the Floyd-Warshall Algorithm
2. Compare it with Dijkstra's Algorithm

All Source Shortest Path

Computes shortest path from any node to any node

Floyd-Warshall's Algorithm is an alternative to Dijkstra in the presence of negative-weight edges (and detect negative weight cycles).

The algorithm begins with shortest paths from existing edges and then computes shortest paths via other nodes. Nodes are incrementally included until the entire graph is considered.

Floyd-Warshall Algorithm

```
1   FloydWarshall(G):
2     Let d be a adj. matrix initialized to +inf
3     foreach (Vertex v : G):
4       d[v][v] = 0
5       s[v][v] = v
6     foreach (Edge (u, v) : G):
7       d[u][v] = cost(u, v)
8       s[u][v] = v
9
10    foreach (Vertex via : G):
11      foreach (Vertex start : G):
12        foreach (Vertex end : G):
13          if (d[start, end] > d[start, via] + d[via, end])
14            d[start, end] = d[start, via] + d[via, end]
15            s[start][end] = s[start][via]
16
17      foreach (Vertex diag: G):
18        if( d[diag][diag] < 0) Negative Cycle Detected
```

Shortest Path Algorithms

Dijkstra's

Graph Assumptions
- Connected, Weighted
- Non-negative Edge weights

Runtime
- Fibonacci Heap: O(n log n + m)
- Binary Heap: O(m log n)

Application
- Single Source Shortest Path

Floyd-Warshall's

Graph Assumptions
- Connected, Weighted
- Negative Edge weights

Runtime
- $O(n^3)$

Application
- All Source Shortest Path
- Detecting Negative Cycles