Data Structures and Algorithms

Agenda

- Bellman Ford
- Warshall Floyd
- Johnson's Algorithm
- A* Search

Graph Distance Algorithms

- Single Source Shortest Path Algorithm
    - BFS based (non-weighted graph)
    - Dijkstra's Algorithm
    - Bellman Ford
- All Pair Shortest Path Algorithm
    - Warshall Floyd Algorithm
    - Johnson's Algorithm
- Point to Point to Shortest Path Algorithm
    - A* Search

Dijkstra's Algorithm

- Single Source Shortest Path Algorithm
- Update the weights of all neighbors

```
if(dist[u] + adjmat[u][v] < dist[v])
    dist[v] = dist[u] + adjmat[u][v];
```

- Greedy programming
- Time: O(V log V)
- Limitation: Cannot work with -ve weighted edges

Dynamic Programming

- DP solution can be applied only if problem has two characteristics
    - Overlapping sub-problems
    - Optimal sub-structures
- DP < Memoization < Recursion

Bellman Ford

- Single Source Shortest Path Algorithm
- Can work with +ve as well as -ve weighted edges
- The -ve weight edges

    - Chemical reaction (energy released +ve or energy absorbed -ve)
    - Electronic circuit (current flowing between two points -- direction)
- Dynamic programming
- Time: O(VE)
- Limitation: Cannot work with -ve weight cycles

All Pair Shortest Path Algorithms

- Single Source Shortest Path Algorithm for all vertices.
    - Dijkstra for all vertices: O(V * V * log V)
        - Cannot work with -ve weight edges
    - Bellman Ford for all vertices: O(V * V * E)
- Output of Single Source Shortest Path Algorithm
    - 1-D Array to store Distance of all vertices from the given vertex.
- Output of All Pair Shortest Path Algorithm
    - 2-D Array to store Distance of all vertices from all vertices.

Warshall Floyd Algorithm

- Time: O(V * V * V)
    - Slower than Dijkstra
    - Faster than Bellman Ford
- Can work with +ve as well as -ve weight edgses

Johnson's Algorithm

A* Search