

Bellman Ford and SPFA

Dr. Mattox Beckman

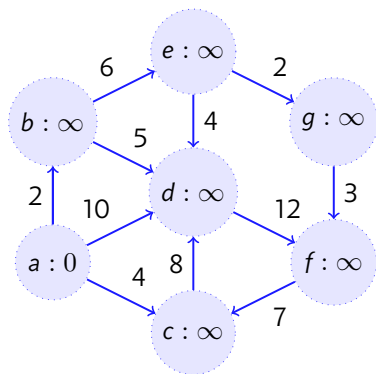
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
DEPARTMENT OF COMPUTER SCIENCE

Objectives

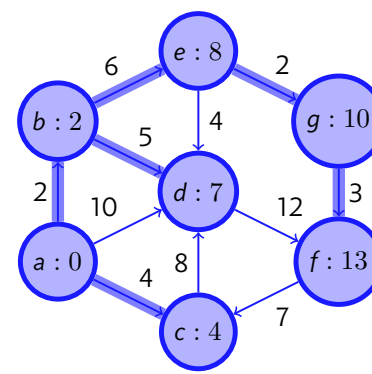
Your Objectives:

- ▶ Explain why Dijkstra's algorithm fails in the presence of negative-weight cycles.
- ▶ Implement SSSP using the Bellman Ford Algorithm
- ▶ Improve over Bellman Ford using the Shortest Path Faster Algorithm

Normal Dijkstra

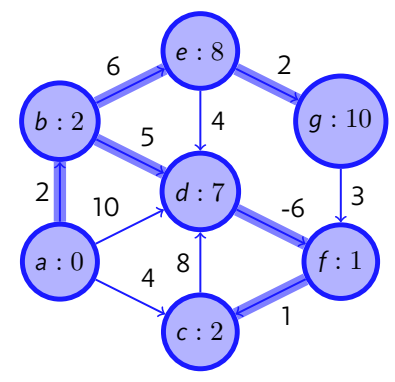
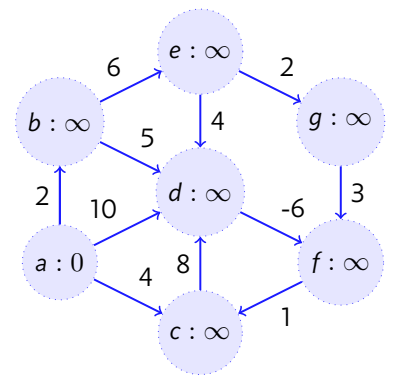


Normal Dijkstra



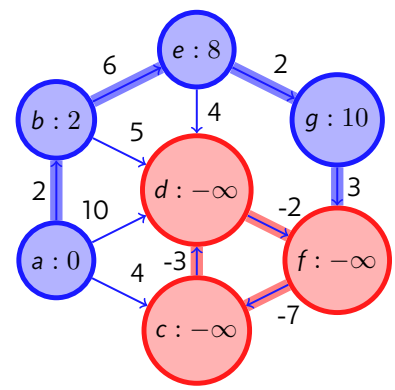
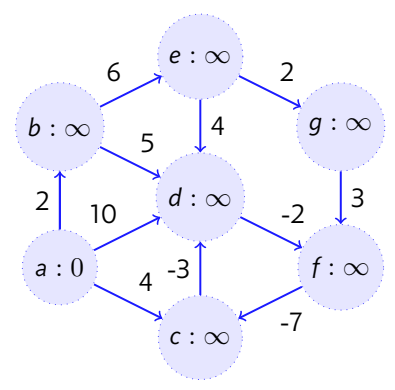
Dijkstra with a Negative Edge

Dijkstra with a Negative Edge



Dijkstra with a Negative Cycle

Dijkstra with a Negative Cycle



Bellman Ford Implementation

```
0 vi dist(V, INF); dist[s] = 0;
1 for (int i = 0; i < V - 1; i++)
2   for (int u = 0; u < V; u++)
3     // these two loops = O(E), overall O(VE)
4     for (int j = 0; j < (int)AdjList[u].size(); j++) {
5       ii v = AdjList[u][j];
6       dist[v.first] = min(dist[v.first], dist[u] + v.second);
7 }
```

Shortest Path Faster Algorithm

```
0 // Credit: Competitive Programming 3
1
2 vi dist(n, INF); dist[S] = 0;
3 queue<int> q; q.push(S);
4 vi in_queue(n, 0); in_queue[S] = 1;
5 while (!q.empty()) {
6   int u = q.front(); q.pop(); in_queue[u] = 0;
7   for (j = 0; j < (int)AdjList[u].size(); j++) {
8     int v = AdjList[u][j].first, weight_u_v = AdjList[u][j].second;
9     if (dist[u] + weight_u_v < dist[v]) {
10      dist[v] = dist[u] + weight_u_v;
11      if (!in_queue[v]) {
12        q.push(v);
13        in_queue[v] = 1;
14 } } } }
```