Bellman Ford and SPFA

Dr. Mattox Beckman

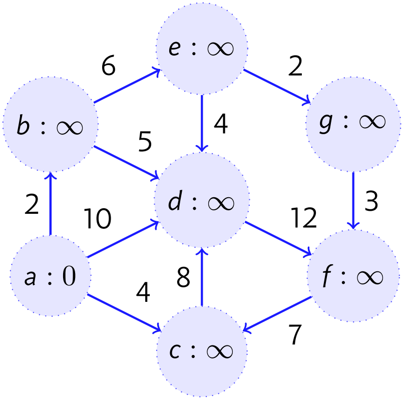
University of Illinois at Urbana-Champaign
Department of Computer Science

Objectives

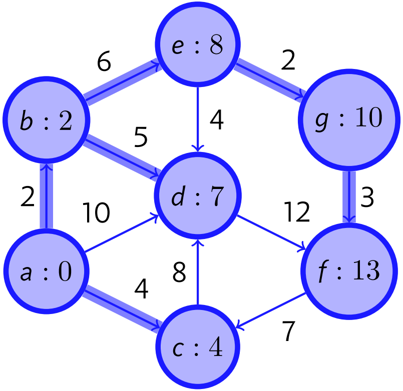
Your Objectives:

- 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

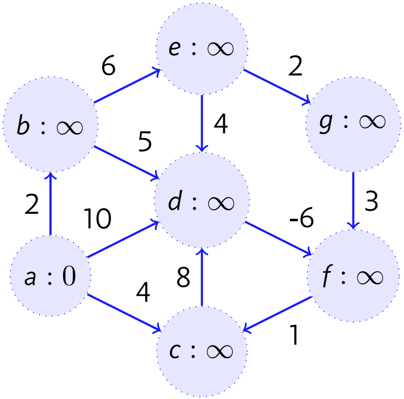
Normal Dijkstra



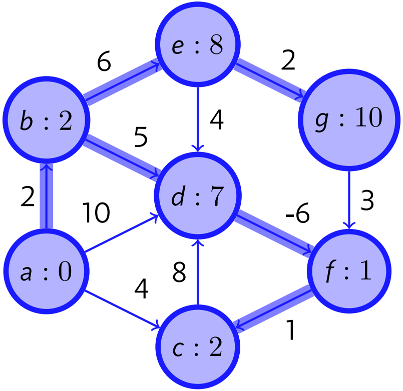
Normal Dijkstra



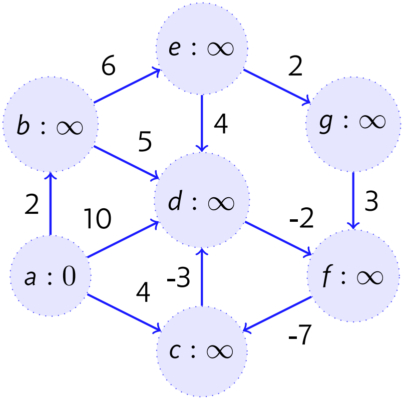
Dijkstra with a Negative Edge



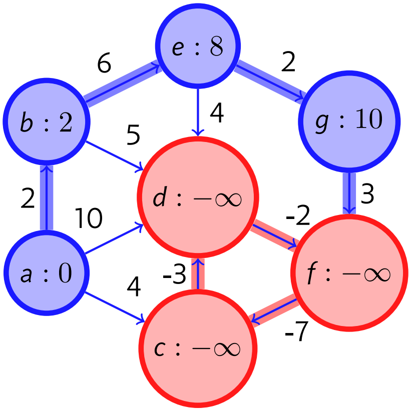
Dijkstra with a Negative Edge



Dijkstra with a Negative Cycle



Dijkstra with a Negative Cycle



Bellman Ford Implementation

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

Shortest Path Faster Algorithm

```
// Credit: Competitive Programming 3

vi dist(n, INF); dist[S] = 0;
queue<int> q; q.push(S);
vi in_queue(n, 0); in_queue[S] = 1;
while (!q.empty()) {
   int u = q.front(); q.pop(); in_queue[u] = 0;
   for (j = 0; j < (int)AdjList[u].size(); j++) {
      int v = AdjList[u][j].first, weight_u_v = AdjList[u][j].second;
      if (dist[u] + weight_u_v < dist[v]) {
         dist[v] = dist[u] + weight_u_v;
         if (!in_queue[v]) {
            q.push(v);
            in_queue[v] = 1;
} } } }
```