Dijkstra's Algorithm

Dr. Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science

Objectives

Your Objectives:

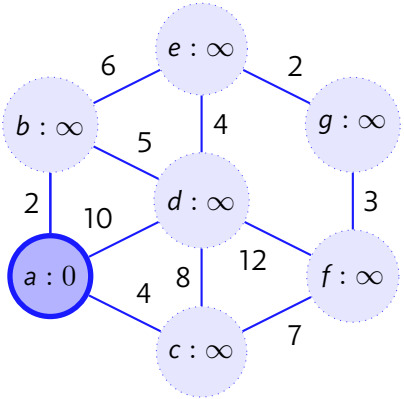
- Implement SSSP using Dijkstra's Algorithm

The Algorithm

- Use this if your graph is weighted.
- Create a distance array and a parent array
- Use a priority queue

	a	b	c	d	e	f	g
Dist	0	∞	∞	∞	∞	∞	∞
Parent	-1	-1	-1	-1	-1	-1	-1

PQueue: a/0

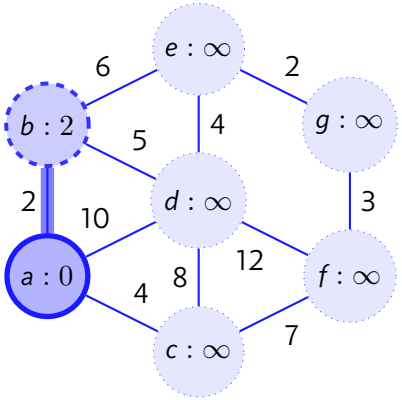


The Algorithm

- Use this if your graph is weighted.
- Create a distance array and a parent array
- Use a priority queue

	a	b	c	d	e	f	g
Dist	0	2	∞	∞	∞	∞	∞
Parent	-1	a	-1	-1	-1	-1	-1

PQueue: b/2

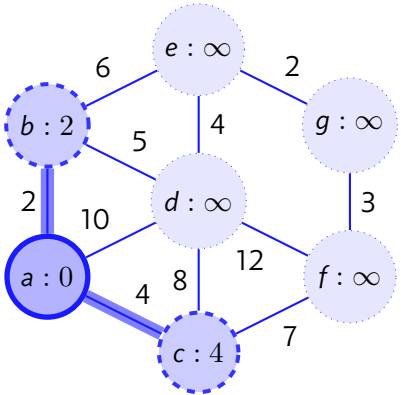


The Algorithm

- Use this if your graph is weighted.
- Create a distance array and a parent array
- Use a priority queue

	a	b	c	d	e	f	g
Dist	0	2	4	∞	∞	∞	∞
Parent	-1	a	a	-1	-1	-1	-1

PQueue: `b/2, c/4`

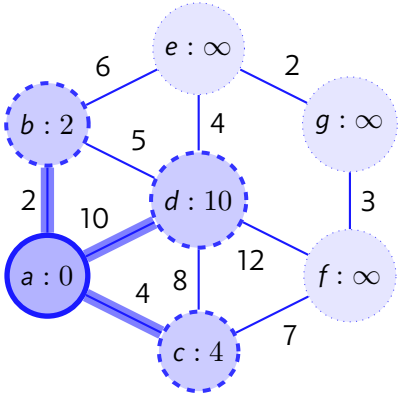


The Algorithm

- Use this if your graph is weighted.
- Create a distance array and a parent array
- Use a priority queue

	a	b	c	d	e	f	g
Dist	0	2	4	10	∞	∞	∞
Parent	-1	a	a	a	-1	-1	-1

PQueue: `b/2, c/4, d/10`

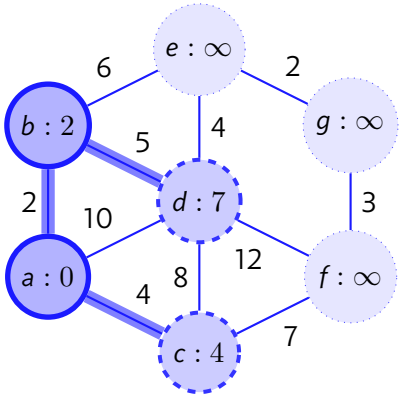


The Algorithm

- Use this if your graph is weighted.
- Create a distance array and a parent array
- Use a priority queue

	a	b	c	d	e	f	g
Dist	0	2	4	7	∞	∞	∞
Parent	-1	a	a	b	-1	-1	-1

PQueue: `c/4, d/7, d/10`

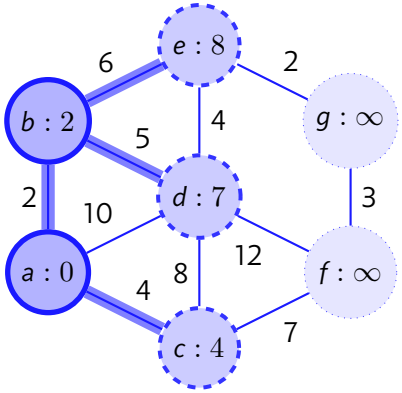


The Algorithm

- Use this if your graph is weighted.
- Create a distance array and a parent array
- Use a priority queue

	a	b	c	d	e	f	g
Dist	0	2	4	7	8	∞	∞
Parent	-1	a	a	b	b	-1	-1

PQueue: `c/4, d/7, e/8, d/10`

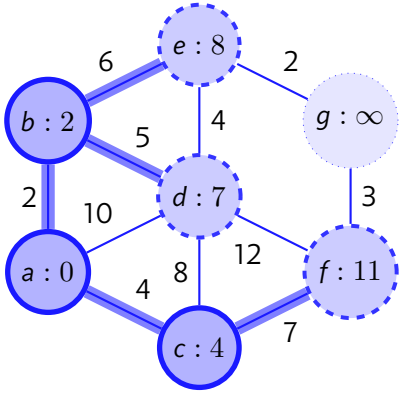


The Algorithm

- Use this if your graph is weighted.
- Create a distance array and a parent array
- Use a priority queue

	a	b	c	d	e	f	g
Dist	0	2	4	7	8	11	∞
Parent	-1	a	a	b	b	c	-1

PQueue: `d/7, e/8, d/10, f/11`

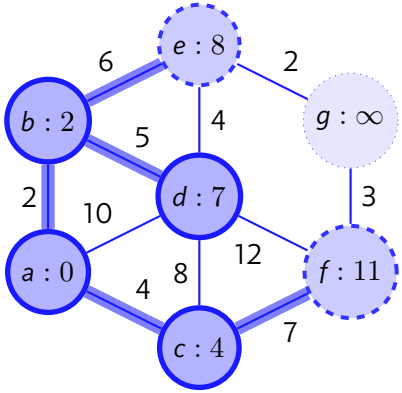


The Algorithm

- Use this if your graph is weighted.
- Create a distance array and a parent array
- Use a priority queue

	a	b	c	d	e	f	g
Dist	0	2	4	7	8	11	∞
Parent	-1	a	a	b	b	c	-1

PQueue: `e/8, d/10, f/11`

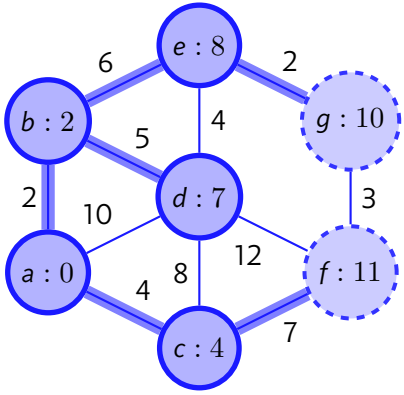


The Algorithm

- Use this if your graph is weighted.
- Create a distance array and a parent array
- Use a priority queue

	a	b	c	d	e	f	g
Dist	0	2	4	7	8	11	10
Parent	-1	a	a	b	b	c	e

PQueue: `d/10, g/10, f/11`

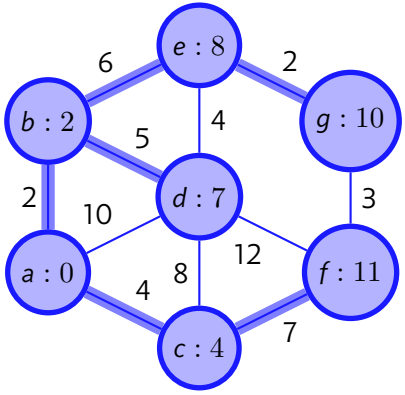


The Algorithm

- Use this if your graph is weighted.
- Create a distance array and a parent array
- Use a priority queue

	a	b	c	d	e	f	g
Dist	0	2	4	7	8	11	10
Parent	-1	a	a	b	b	c	e

PQueue: Done!



Implementation

```
0 // Credit: Competitive Programming 3
1 vi dist(V, INF); dist[s] = 0;
2 priority_queue< ii, vector<ii>, greater<ii> > pq; pq.push(ii(0, s));
3 while (!pq.empty()) {
4    ii front = pq.top(); pq.pop(); // get shortest unvisited vertex
5    int d = front.first, u = front.second;
6    if (d > dist[u]) continue; // lazy delete
7    for (int j = 0; j < (int)AdjList[u].size(); j++) {
8       ii v = AdjList[u][j];
9       if (dist[u] + v.second < dist[v.first]) {
10         dist[v.first] = dist[u] + v.second; // relax operation
11         pq.push(ii(dist[v.first], v.first));
12 } } }
```