



CSE332: Data Abstractions

Section 10

Hyeln Kim

Winter 2014

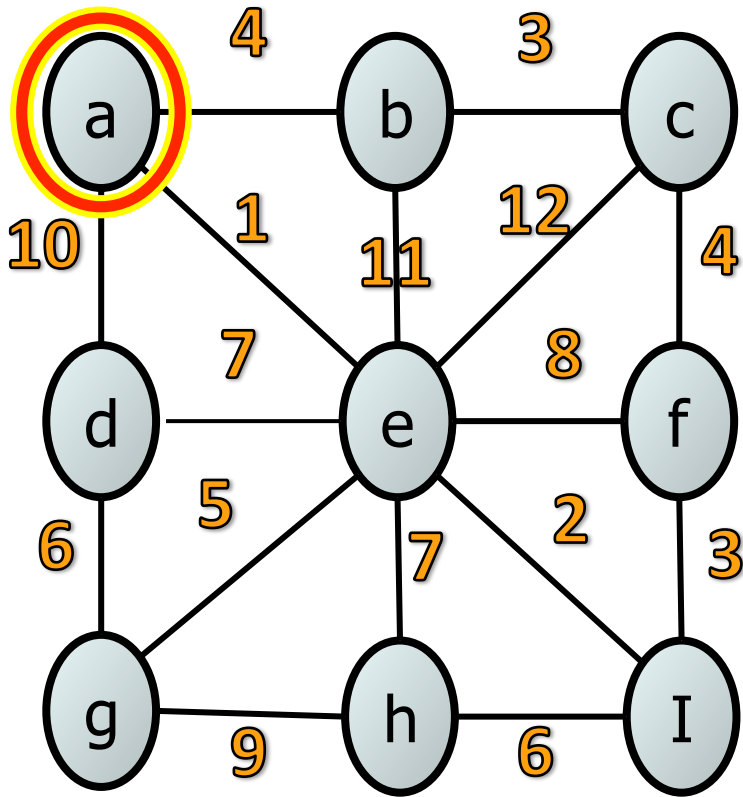
Find Shortest Path

Dijkstra's Algorithm

Dijkstra's Algorithm

Source Node: A

Pick one with shortest distance from source: **A**

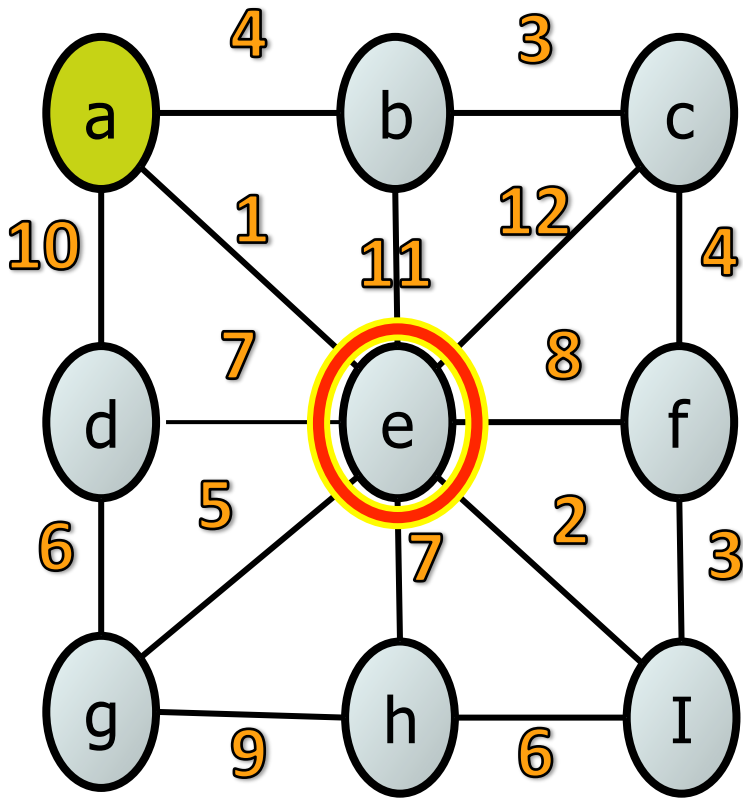


Node	Mark	Dist	Path	Mark	Dist	Path
A		0		1	0	-
B		∞			4	A
C		∞			∞	
D		∞			10	A
E		∞			1	A
F		∞			∞	
G		∞			∞	
H		∞			∞	
I		∞			∞	

Dijkstra's Algorithm

Source Node: A

Pick one with shortest distance from source: **E**

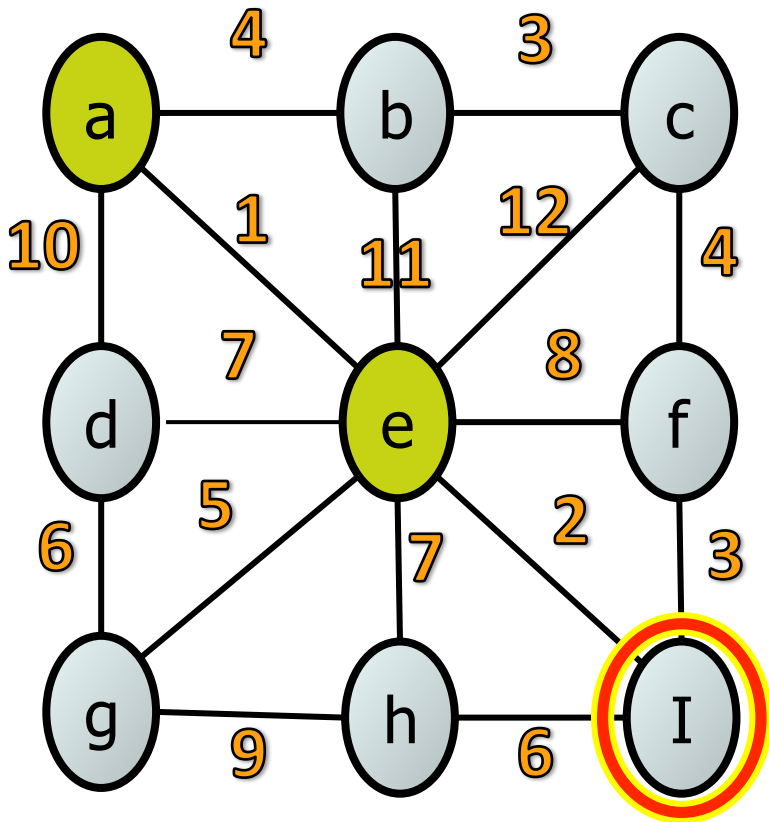


Node	Mark	Dist	Path	Mark	Dist	Path
A	1	0	-	1	0	-
B		4	A		4	A
C		∞			13	E
D		10	A		8	E
E		1	A	1	1	A
F		∞			9	E
G		∞			6	E
H		∞			8	E
I		∞			3	E

Dijkstra's Algorithm

Source Node: A

Pick one with shortest distance from source: **I**

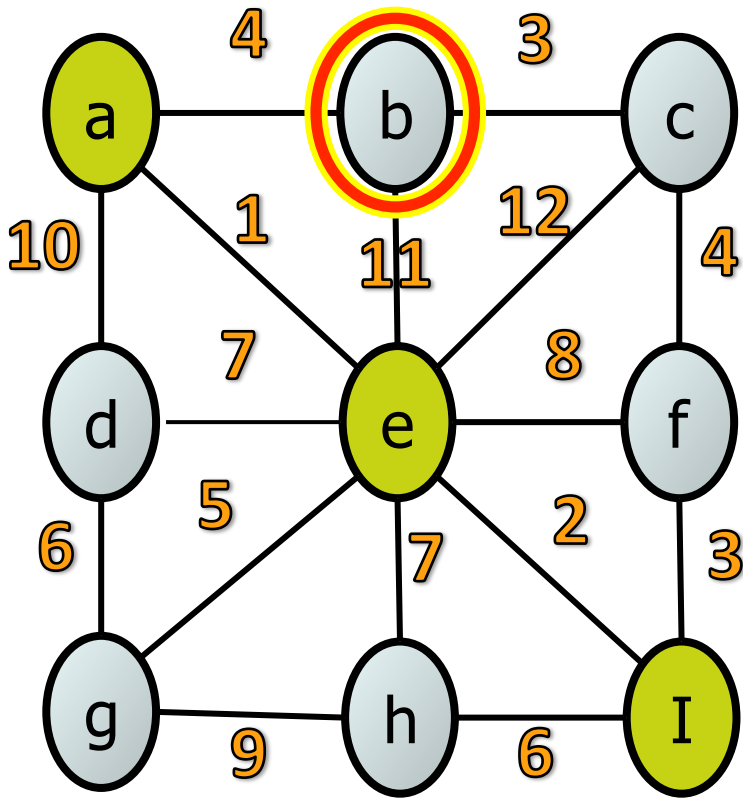


Node	Mark	Dist	Path	Mark	Dist	Path
A	1	0	-	1	0	-
B		4	A		4	A
C		13	E		13	E
D		8	E		8	E
E	1	1	A	1	1	A
F		9	E		6	I
G		6	E		6	E
H		8	E		8	E
I		3	E	1	3	E

Dijkstra's Algorithm

Source Node: A

Pick one with shortest distance from source: **B**

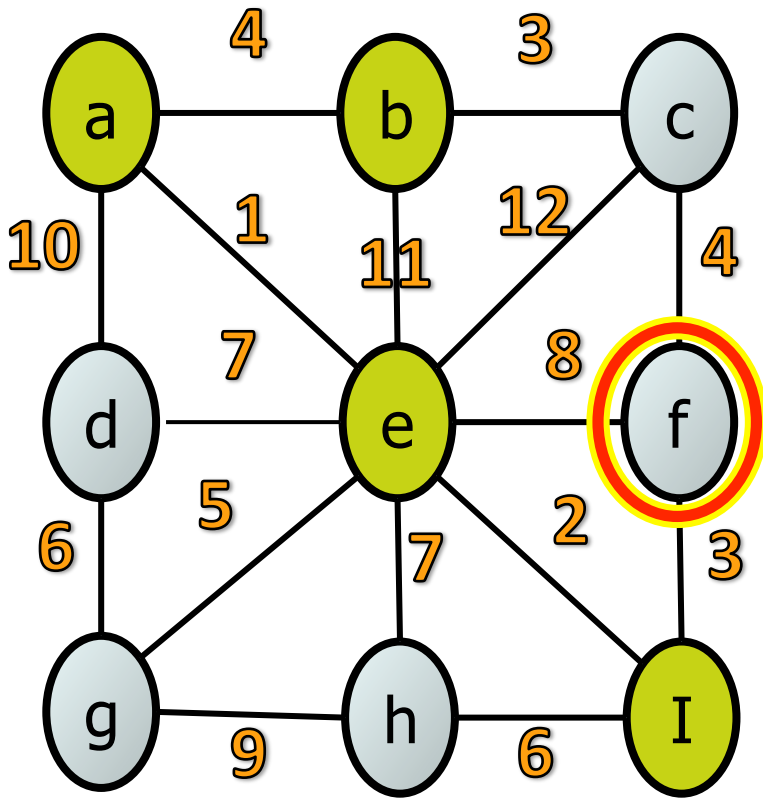


Node	Mark	Dist	Path	Mark	Dist	Path
A	1	0	-	1	0	-
B		4	A	1	4	A
C		13	E		7	B
D		8	E		8	E
E	1	1	A	1	1	A
F		6	I		6	I
G		6	E		6	E
H		8	E		8	E
I	1	3	E	1	3	E

Dijkstra's Algorithm

Source Node: A

Pick one with shortest distance from source: **F**

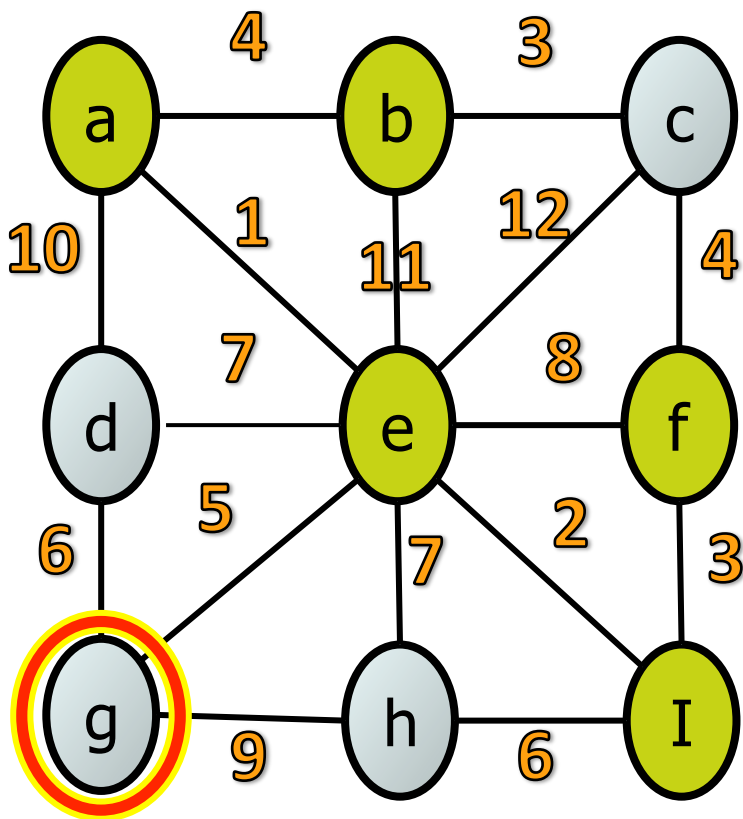


Node	Mark	Dist	Path	Mark	Dist	Path
A	1	0	-	1	0	-
B	1	4	A	1	4	A
C		7	B		7	B
D		8	E		8	E
E	1	1	A	1	1	A
F		6	I	1	6	I
G		6	E		6	E
H		8	E		8	E
I	1	3	E	1	3	E

Dijkstra's Algorithm

Source Node: A

Pick one with shortest distance from source: **G**

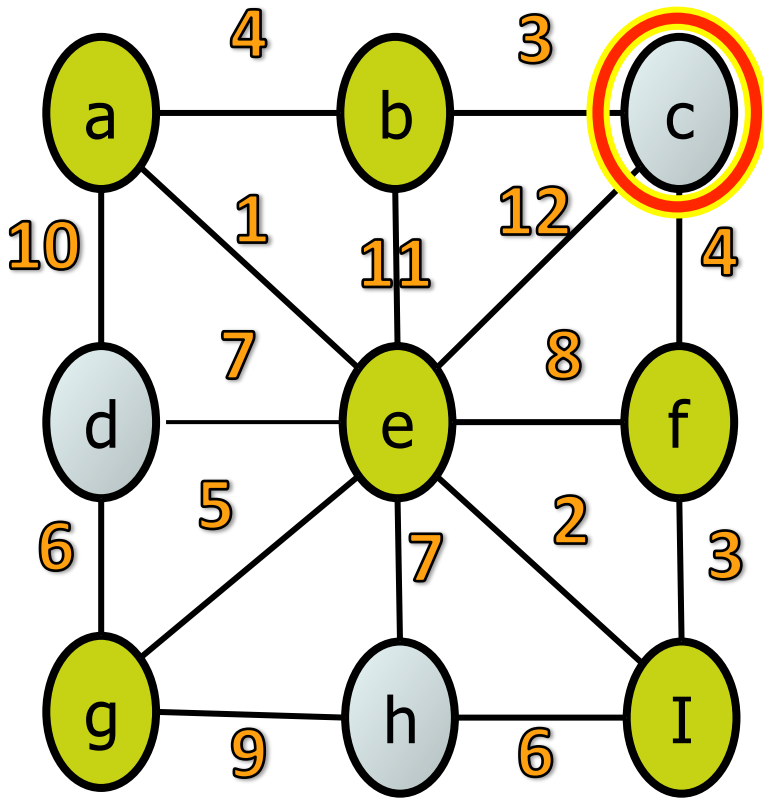


Node	Mark	Dist	Path	Mark	Dist	Path
A	1	0	-	1	0	-
B	1	4	A	1	4	A
C		7	B		7	B
D		8	E		8	E
E	1	1	A	1	1	A
F	1	6	I	1	6	I
G		6	E	1	6	E
H		8	E		8	E
I	1	3	E	1	3	E

Dijkstra's Algorithm

Source Node: A

Pick one with shortest distance from source: **C**

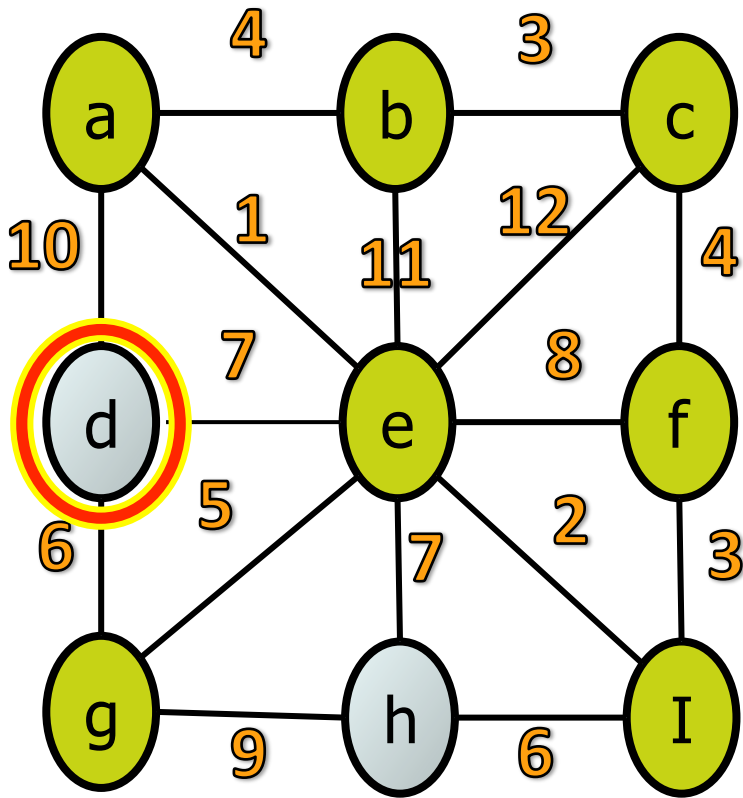


Node	Mark	Dist	Path	Mark	Dist	Path
A	1	0	-	1	0	-
B	1	4	A	1	4	A
C		7	B	1	7	B
D		8	E		8	E
E	1	1	A	1	1	A
F	1	6	I	1	6	I
G	1	6	E	1	6	E
H		8	E		8	E
I	1	3	E	1	3	E

Dijkstra's Algorithm

Source Node: A

Pick one with shortest distance from source: **D**

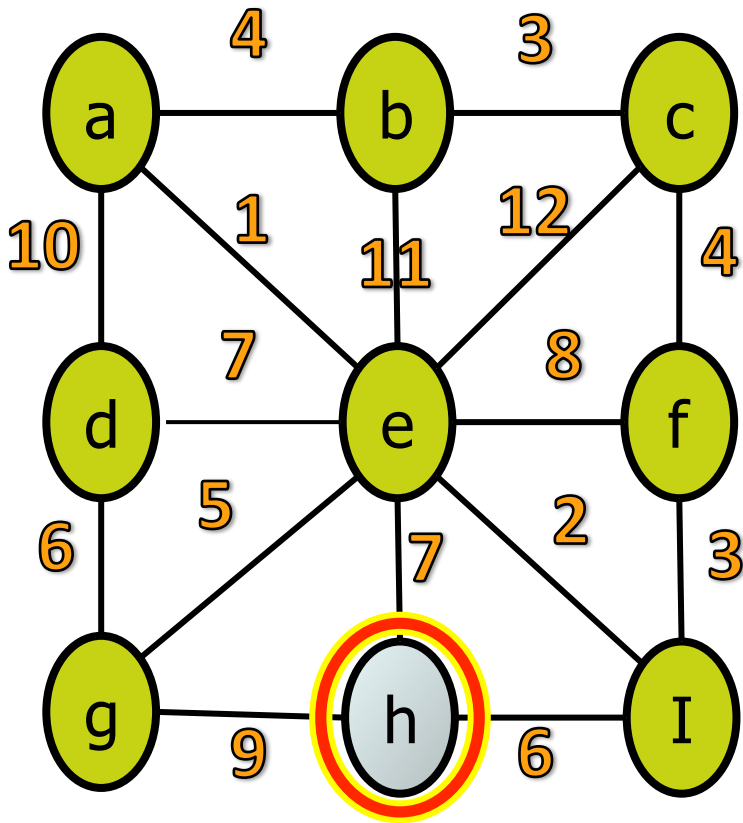


Node	Mark	Dist	Path	Mark	Dist	Path
A	1	0	-	1	0	-
B	1	4	A	1	4	A
C	1	7	B	1	7	B
D		8	E	1	8	E
E	1	1	A	1	1	A
F	1	6	I	1	6	I
G	1	6	E	1	6	E
H		8	E		8	E
I	1	3	E	1	3	E

Dijkstra's Algorithm

Source Node: A

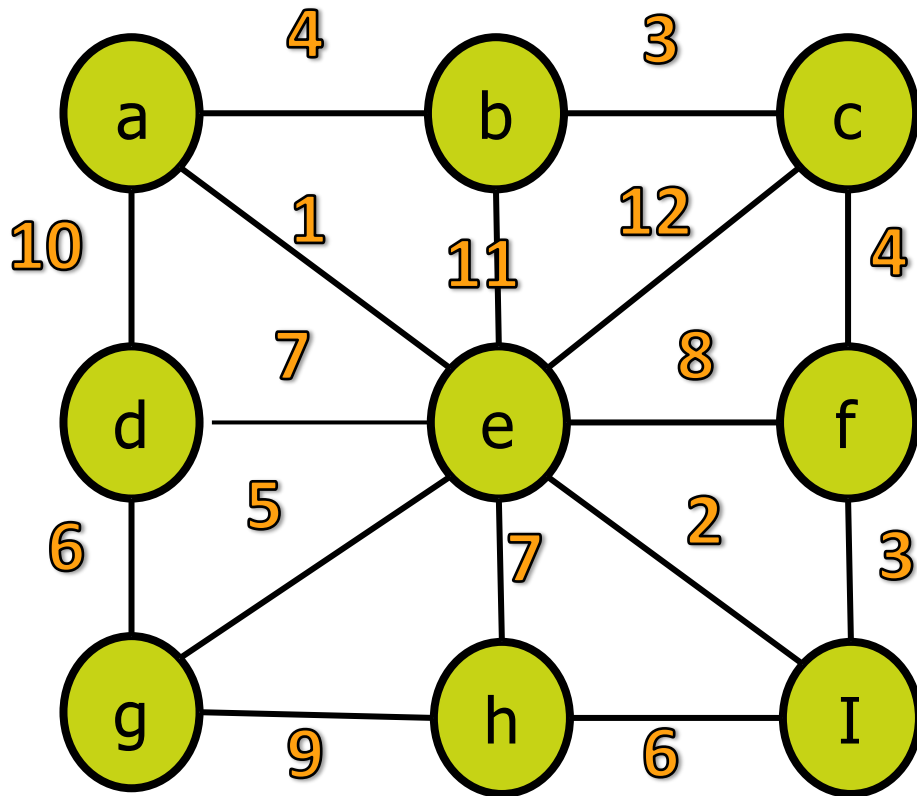
Pick one with shortest distance from source: **H**



Node	Mark	Dist	Path	Mark	Dist	Path
A	1	0	-	1	0	-
B	1	4	A	1	4	A
C	1	7	B	1	7	B
D	1	8	E	1	8	E
E	1	1	A	1	1	A
F	1	6	I	1	6	I
G	1	6	E	1	6	E
H		8	E	1	8	E
I	1	3	E	1	3	E

Dijkstra's Algorithm

Source Node: A

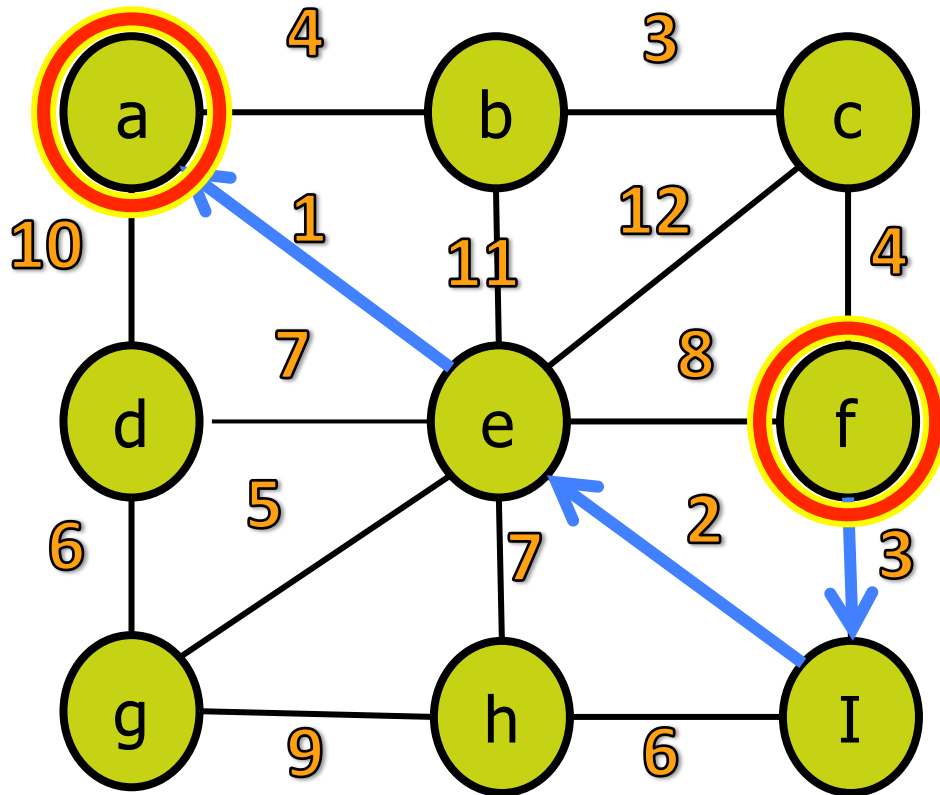


Done!

Node	Mark	Dist	Path
A	1	0	-
B	1	4	A
C	1	7	B
D	1	8	E
E	1	1	A
F	1	6	I
G	1	6	E
H	1	8	E
I	1	3	E

Dijkstra's Algorithm

Source Node: A



Find shortest path from F to A

Node	Mark	Dist	Path
A	1	0	-
B	1	4	A
C	1	7	B
D	1	8	E
E	1	1	A
F	1	6	I
G	1	6	E
H	1	8	E
I	1	3	E

Dijkstra's Algorithm

- **Dijkstra's Algorithm Runtime**

- Initializing each node $O(|V|)$

- Pick smallest v & Mark v

- Single step (No PQ / PQ): $O(|V|)$ $O(\log |V|)$

- Total (No PQ / PQ): $O(|V|^2)$ $O(|V| \cdot \log |V|)$

- Update cost of all neighbors of v

- Total (No PQ): $O(|E|)$

- Total (PQ): $O(|E| \cdot \log |V|)$

- **Total Runtime:** $O(|V|^2 + |E|)$ No Priority Queue

- $O((|V| + |E|) \cdot \log |V|)$ Priority Queue

Dijkstra's Algorithm

- **Total Runtime:** $O(|V|^2 + |E|)$ No Priority Queue
 $O((|V| + |E|) * \log |V|)$ Priority Queue
- Sparse graph: $|V| \gg \gg |E|$, $O(|V| * \log |V|)$
Better with Priority Queue
- Dense graph: $|E| \gg \gg |V|$, $O(|E| * \log |V|)$
 $\Rightarrow O(|V|^2 * \log |V|)$
Better without Priority Queue