



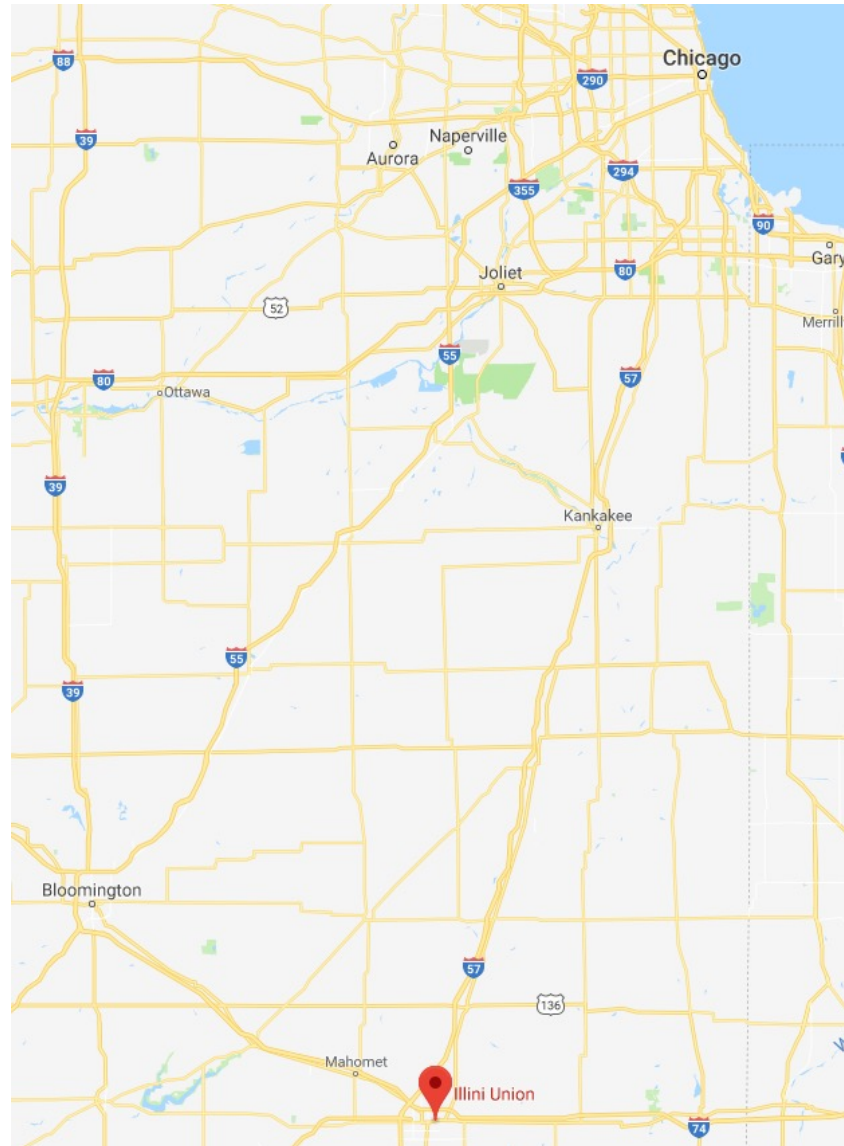
CS 225

Data Structures

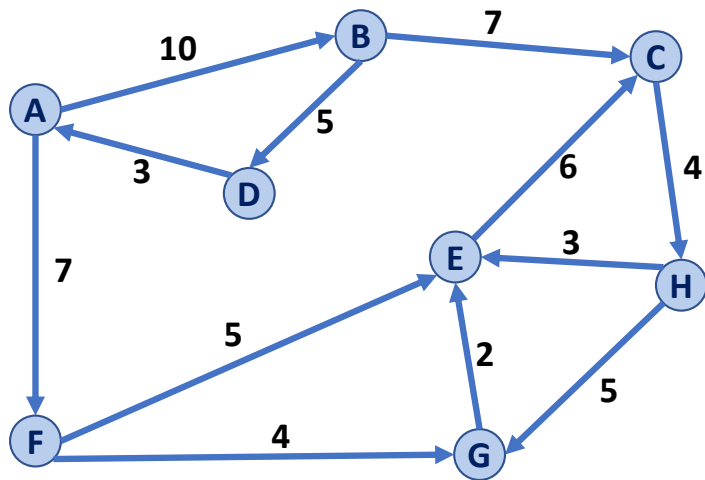
April 1 – Dijkstra's Algorithm

G Carl Evans

Shortest Path



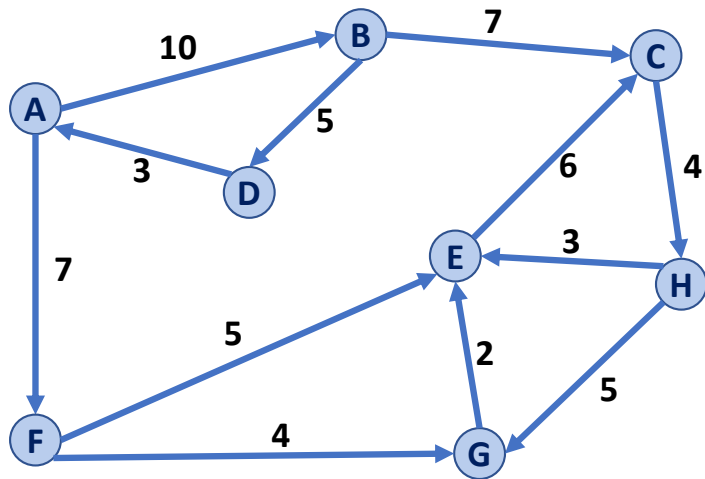
Dijkstra's Algorithm (SSSP)



```
PrimMST(G, s):
```

```
6  foreach (Vertex v : G):
7    d[v] = +inf
8    p[v] = NULL
9    d[s] = 0
10
11  PriorityQueue Q // min distance, defined by d[v]
12  Q.buildHeap(G.vertices())
13
14  repeat n times:
15    Vertex u = Q.removeMin()
16    foreach (Vertex v : neighbors of u not in T):
17      if cost(u, v) < d[v]:
18        d[v] = cost(u, v)
19        p[v] = u
20
21
```

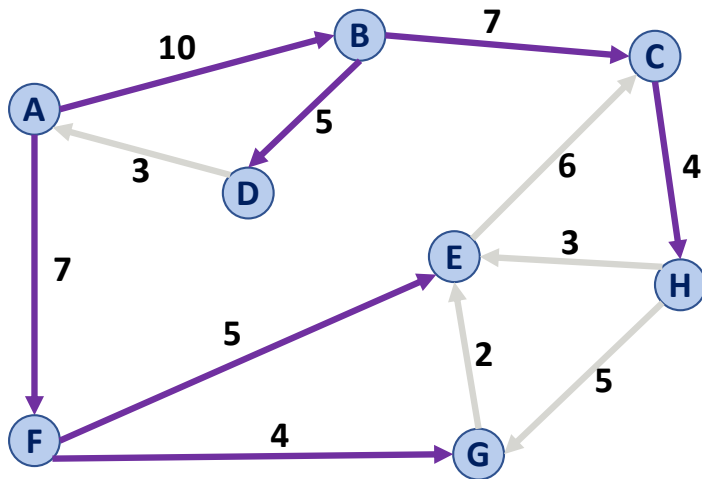
Dijkstra's Algorithm (SSSP)



```
DijkstraSSSP(G, s):
```

```
6  foreach (Vertex v : G):
7      d[v] = +inf
8      p[v] = NULL
9      d[s] = 0
10
11  PriorityQueue Q // min distance, defined by d[v]
12  Q.buildHeap(G.vertices())
13  Graph T          // "labeled set"
14
15  repeat n times:
16      Vertex u = Q.removeMin()
17      T.add(u)
18      foreach (Vertex v : neighbors of u not in T):
19          if _____ < d[v]:
20              d[v] = _____
21              p[v] = u
```

Dijkstra's Algorithm (SSSP)

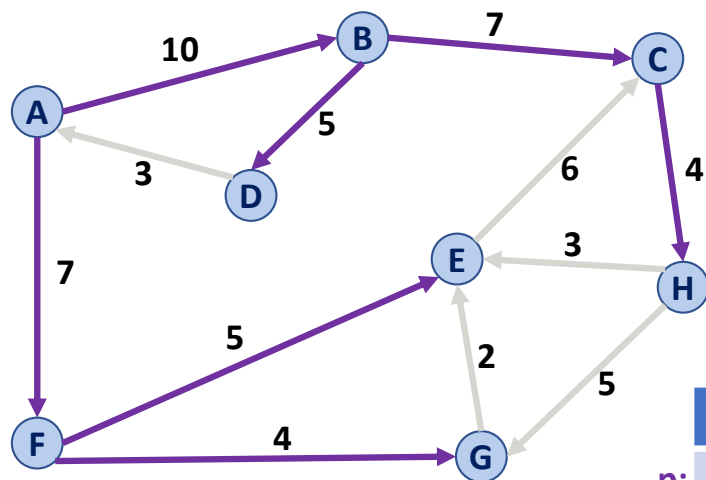


```
DijkstraSSSP(G, s):
```

```
6  foreach (Vertex v : G):
7     d[v] = +inf
8     p[v] = NULL
9     d[s] = 0
10
11  PriorityQueue Q // min distance, defined by d[v]
12  Q.buildHeap(G.vertices())
13
14  repeat n times:
15     Vertex u = Q.removeMin()
16     foreach (Vertex v : neighbors of u not in T):
17         if cost(u, v) + d[u] < d[v]:
18             d[v] = cost(u, v) + d[u]
19             p[v] = u
20
21
```

Dijkstra's Algorithm (SSSP)

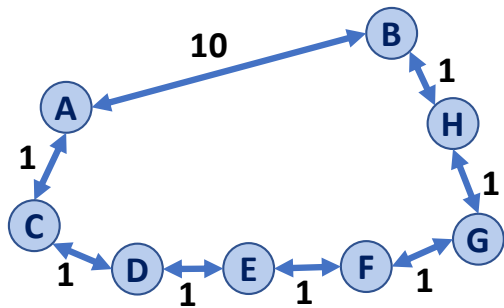
Dijkstra gives us the shortest path from our path (single source) to **every** connected vertex!



	A	B	C	D	E	F	G	H
p:	--	A	B	B	F	A	F	C
d:	0	10	17	15	12	7	11	21

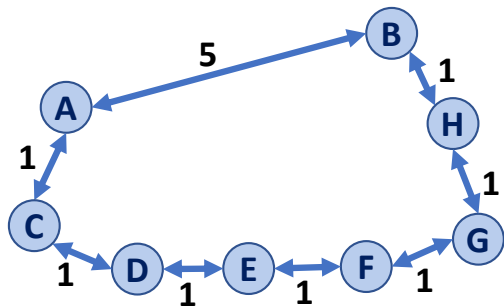
Dijkstra's Algorithm (SSSP)

Q: How does Dijkstra handle a single heavy-weight path vs. many light-weight paths?



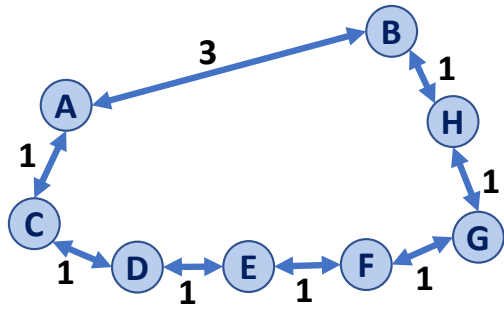
Dijkstra's Algorithm (SSSP)

Q: How does Dijkstra handle a single heavy-weight path vs. many light-weight paths?



Dijkstra's Algorithm (SSSP)

Q: How does Dijkstra handle undirected graphs?



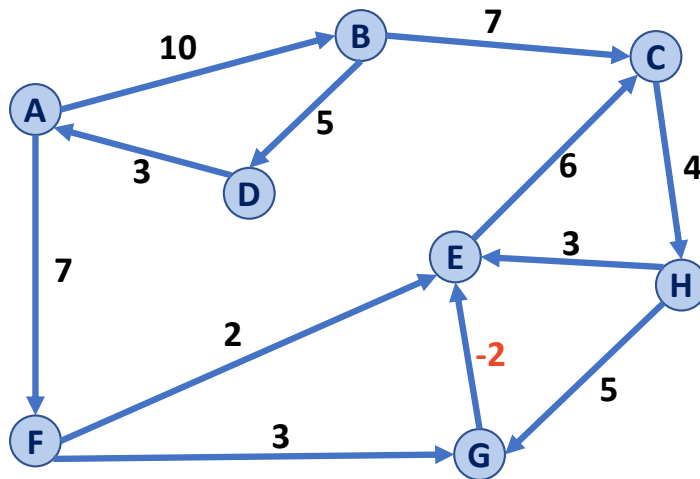
Dijkstra's Algorithm (SSSP)

What is Dijkstra's running time?

```
DijkstraSSSP(G, s):
6   foreach (Vertex v : G):
7       d[v] = +inf
8       p[v] = NULL
9   d[s] = 0
10
11   PriorityQueue Q // min distance, defined by d[v]
12   Q.buildHeap(G.vertices())
13
14   repeat n times:
15       Vertex u = Q.removeMin()
16       foreach (Vertex v : neighbors of u not in T):
17           if cost(u, v) + d[u] < d[v]:
18               d[v] = cost(u, v) + d[u]
19               p[v] = u
20
21   return T
22
23
```

Dijkstra's Algorithm (SSSP)

Q: How does Dijkstra handle negative weight edges?

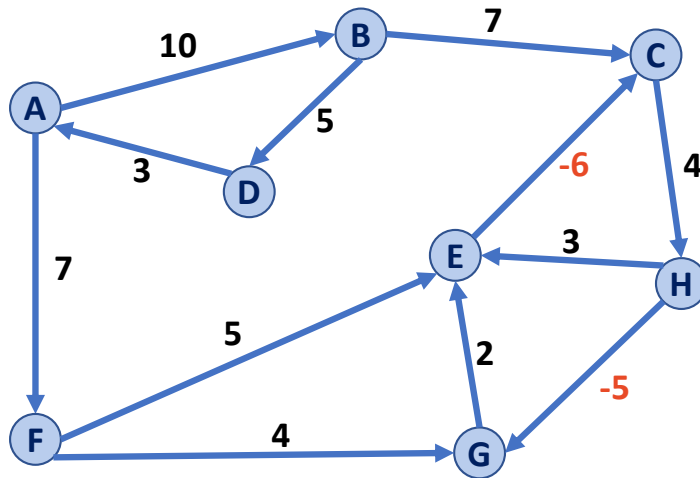


Modified Dijkstra's Algorithm (SSSP)

```
DijkstraSSSP(G, s):
6  foreach (Vertex v : G):
7      d[v] = +inf
8      p[v] = NULL
9  d[s] = 0
10
11  PriorityQueue Q // min distance, defined by d[v]
12  Q.buildHeap(G.vertices())
13  Graph T          // "labeled set"
14
15  repeat until Q.empty() times:
16      Vertex u = Q.removeMin()
17      foreach (Vertex v : neighbors of u not in T):
18          if cost(u, v) + d[u] < d[v]:
19              d[v] = cost(u, v) + d[u]
20              p[v] = u
21              Q.push(v)
22
23  return T
```

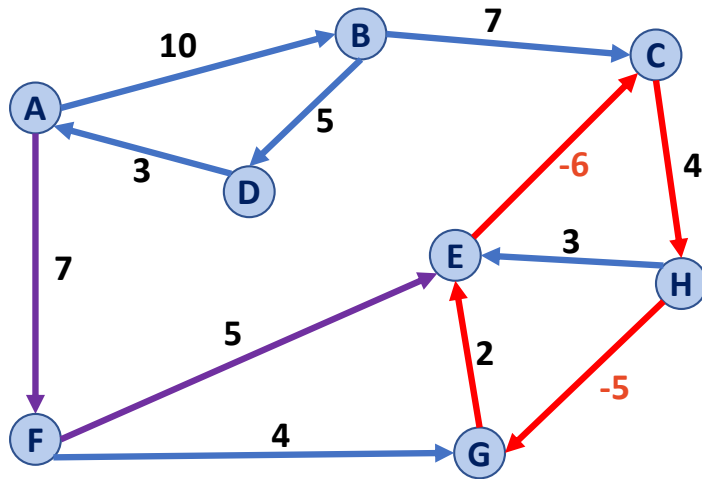
Dijkstra's Algorithm (SSSP)

Q: How does Dijkstra handle negative weight cycles?



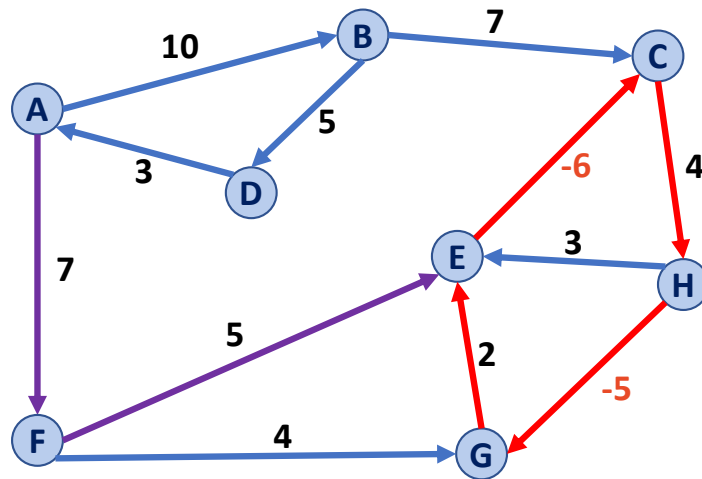
Dijkstra's Algorithm (SSSP)

Q: How does Dijkstra handle negative weight cycles?



Dijkstra's Algorithm (SSSP)

Q: How does Dijkstra handle negative weight cycles?



Shortest Path (A → E): A → F → E → (C → H → G → E)*
Length: 12 Length: -5 (repeatable)