



Single Source Shortest Path

Learning Objectives

1. Implement Dijkstra's Algorithm
2. Know the runtime of Dijkstra's

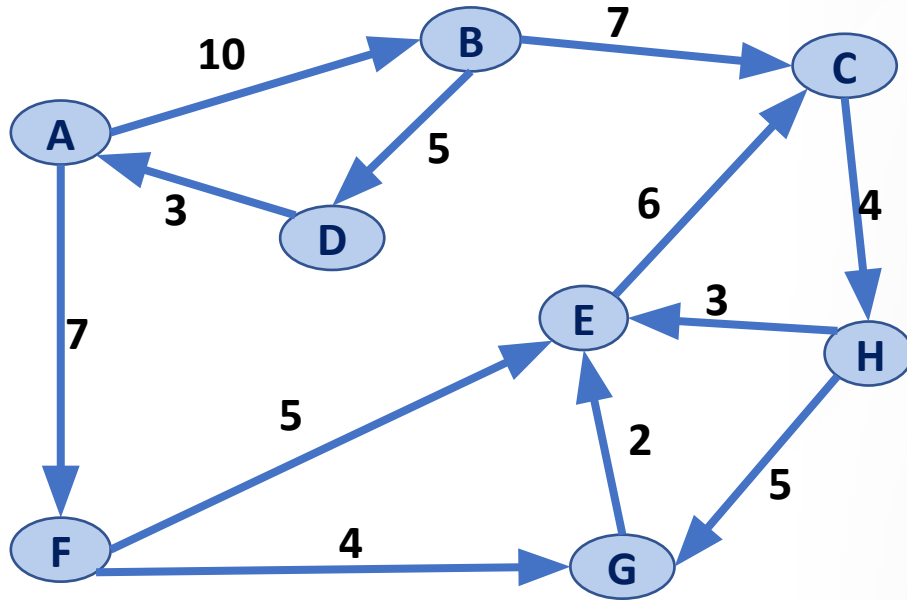




Shortest Path



Example



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



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

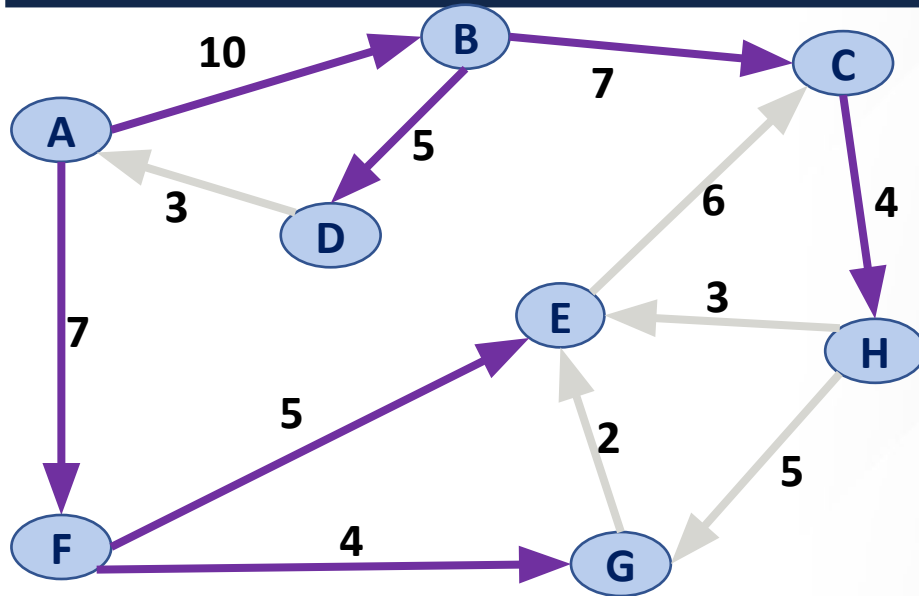


```
1 DijkstraSSSP(G, s):
2   foreach (Vertex v : G):
3     d[v] = +inf
4     p[v] = NULL
5   d[s] = 0
6
7   PriorityQueue Q // min distance, defined by d[v]
8   Q.buildHeap(G.vertices())
9   Graph T          // "labeled set"
10
11  repeat n times:
12    Vertex u = Q.removeMin()
13    T.add(u)
14   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

```
1 PrimMST(G, s):
2   foreach (Vertex v : G):
3     d[v] = +inf
4     p[v] = NULL
5   d[s] = 0
6
7   PriorityQueue Q // min distance, defined by d[v]
8   Q.buildHeap(G.vertices())
9   Graph MST        // "labeled set"
10
11  repeat n times:
12    Vertex m = Q.removeMin()
13    MST.add(m)
14    foreach (Vertex v : neighbors of m not in MST):
15      if cost(v, m) < d[v]:
16        d[v] = cost(v, m)
17        p[v] = m
18
19  return MST
```



Final Result



Runtime

Binary Heap: $O(m \log(n))$

Fibonacci Heap: $O(n \log(n) + m)$

Assumptions: Weighted, Directed

	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

