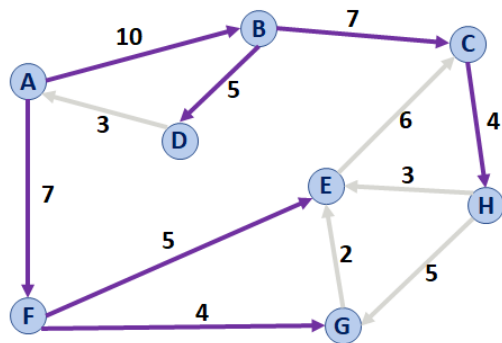


Dijkstra's Algorithm (Single Source Shortest Path)

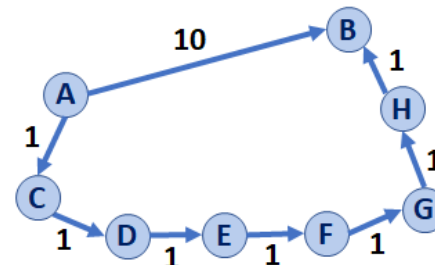


Dijkstra's Algorithm Overview:

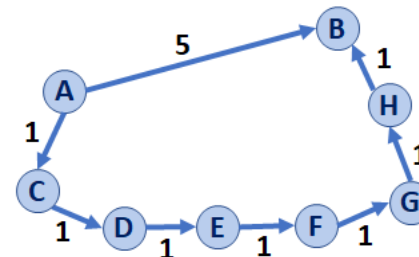
- The overall logic is the same as Prim's Algorithm
- We will modify the code in only two places – both involving the update to the distance metric.
- The result is a directed acyclic graph or DAG

Pseudocode for Dijkstra's SSSP Algorithm	
1	DijkstraSSSP(G, s):
2	Input: G, Graph;
3	s, vertex in G, starting vertex of algorithm
4	Output: T, DAG w/ shortest paths (and distances) to s
5	
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 m = Q.removeMin()
17	T.add(m)
18	foreach (Vertex v : neighbors of m not in T):
19	if $d[u] + \text{cost}(u, v) < d[v]$:
20	d[v] = $d[u] + \text{cost}(u, v)$
21	p[v] = m
22	
23	return T

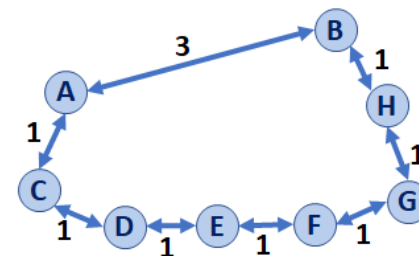
Dijkstra: One heavy-weight edge vs. faster light-weight edges?



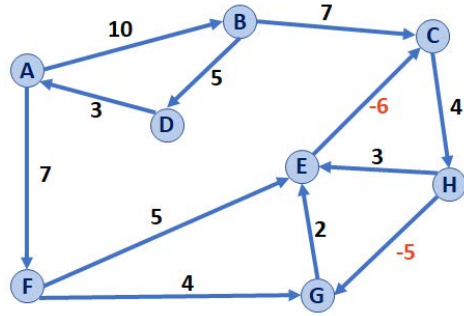
Dijkstra: One medium-weight edge vs. many light-weight edges?



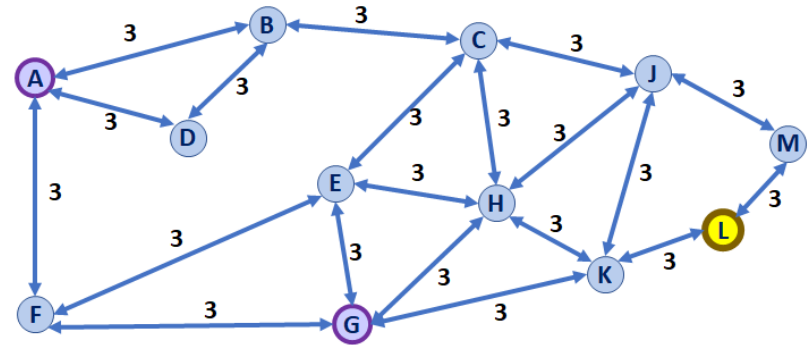
Dijkstra: Undirected graphs?



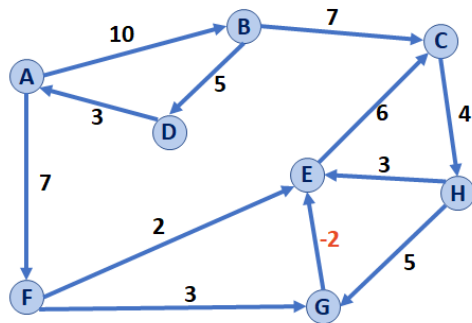
Dijkstra: What if we have a negative-weight cycle?



Landmark Path Problem: My favorite graph problem!



Dijkstra: What if we have a minimum-weight edge, without having a negative-weight cycle?



...what assumption does Dijkstra's algorithm make?

Dijkstra: What is the running time?

CS 225 – Things To Be Doing:

1. Final Exam begins on Reading Day
2. MP7 – Part 3 is released; two-day grace period!
3. lab_finale due Sunday
4. Daily POTDs are ongoing for +1 point /problem