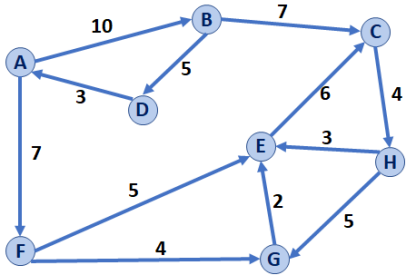
CS 225

#41: Dijkstra's + Shortest Path

December 8, 2017 · *Wade Fagen-Ulmschneider*

Dijkstra's Algorithm (Single Source Shortest Path)



Dijkstra's Algorithm Overview:

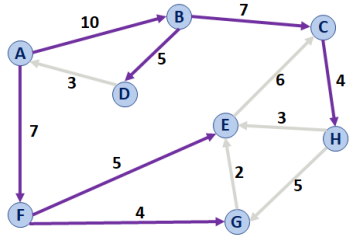
- The overall logic is the same as Prim's Algorithm
- We will modify the code in only two places – both involving the update to the distance metric.
- The result is a directed acyclic graph or DAG

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 with 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 cost(u, v) + d[u] < d[v]:
20         d[v] = cost(u, v) + d[u]
21         p[v] = m
22
23   return T
```

Backtracking in Dijkstra

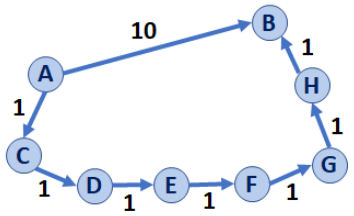
Dijkstra's Algorithm gives us the shortest path from a single source to every connected vertex:



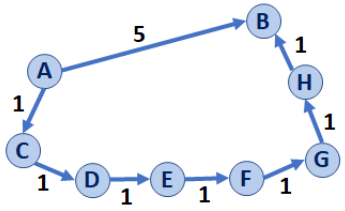
	A	B	C	D	E	F	G	H
p	NULL	A	B	B	F	A	F	C
d	0	10	17	15	12	7	11	21

Examples: How is a single heavy-weight path vs. many light-weight paths handled?

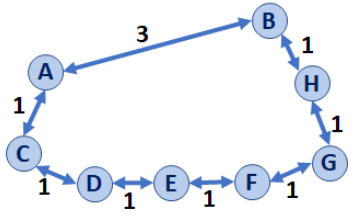
Ex 1:



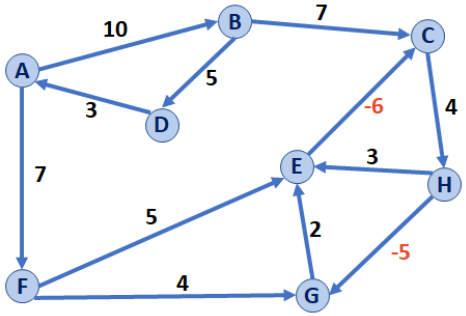
Ex 2:



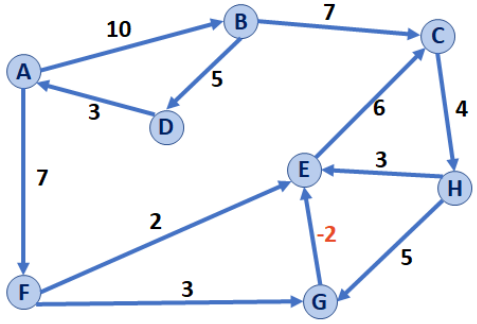
What about undirected graphs?



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



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



Dijkstra makes an assumption:

Dijkstra: What is the running time?

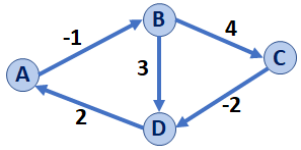
Floyd-Warshall Algorithm

Floyd-Warshall's Algorithm is an alterative to Dijkstra in the presence of negative-weight edges (but not negative weight cycles).

Pseudocode for Floyd-Warshall's Algorithm

```
1  FloydWarshall(G):
2    Input: G, Graph;
3    Output: d, an adjacency matrix of distances between all
4            vertex pairs
5
6    Let d be a adj. matrix initialized to +inf
7    foreach (Vertex v : G):
8      d[v][v] = 0
9    foreach (Edge (u, v) : G):
10     d[u][v] = cost(u, v)
11
12   foreach (Vertex u : G):
13     foreach (Vertex v : G):
14       foreach (Vertex w : G):
15         if d[u, v] > d[u, w] + d[w, v]:
16           d[u, v] = d[u, w] + d[w, v]
17
18   return d
```

Running Floyd-Warshall's Algorithm



CS 225 – Things To Be Doing:

1. Exam #13 (makeup exam) starts Monday
2. MP7 due Monday, Dec. 11 at 11:59pm
3. lab_ml due Sunday, Dec. 10 at 11:59pm
4. Multi-day "puzzle" POTDs available M/W/F