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

Dijkstra’s Algorithm (Single Source Shortest Path)

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