Prim’s Algorithm (Minimum Spanning Tree)

Pseudocode for Prim’s MST Algorithm

```
PrimMST(G, s):
    Input: G, Graph;
    s, vertex in G, starting vertex of algorithm
    Output: T, a minimum spanning tree (MST) of G
    foreach (Vertex v : G):
        d[v] = +inf
        p[v] = NULL
    d[s] = 0
    PriorityQueue Q // min distance, defined by d[v]
    Q.buildHeap(G.vertices())
    Graph T // "labeled set"
    repeat n times:
        Vertex m = Q.removeMin()
        T.add(m)
        foreach (Vertex v : neighbors of m not in T):
            if cost(v, m) < d[v]:
                d[v] = cost(v, m)
                p[v] = m
    return T
```

Running Time of MST Algorithms

- Kruskal’s Algorithm:
- Prim’s Algorithm:

Q: What must be true about the connectivity of a graph when running an MST algorithm?

...what does this imply about the relationship between n and m?

Q: Suppose we built a new heap that optimized the decrease-key operation, where decreasing the value of a key in a heap updates the heap in amortized constant time, or $O(1)^*$. How does that change Prim’s Algorithm runtime?

Shortest Path Home:
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

Dijkstra makes an assumption:

**Pseudocode for Dijkstra’s SSSP Algorithm**

```java
DijkstraSSSP(G, s):
  Input: G, Graph;
  s, vertex in G, starting vertex of algorithm
  Output: T, DAG w/ shortest paths (and distances) to s

  foreach (Vertex v : G):
    d[v] = +inf
    p[v] = NULL
  
  d[s] = 0
  PriorityQueue Q // min distance, defined by d[v]
  Q.buildHeap(G.vertices())
  Graph T // "labeled set"

  repeat n times:
    Vertex m = Q.removeMin()
    T.add(m)
    foreach (Vertex v : neighbors of m not in T):
      if ____________________ < d[v]:
        d[v] = ____________________
        p[v] = m

  return T
```

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?

---

**CS 225 – Things To Be Doing:**

1. Final Exam runs Thursday, May 3 – Thursday, May 10
2. MP7 is released; EC due tonight, Monday, April 23rd
3. Final lab, lab_ml, released Wednesday
4. This week is the last week of POTDs (last POTD is Friday!)