Aditya Shirode, Rhythm Shah, Shrenuj Gandhi

4. [8 points] Suppose you have a weighted, undirected graph G with positive edge weights and a start vertex s. Come up with an algorithm that runs as fast as Dijkstra's and assigns a label usp[u] to every vertex u in G, so that usp[u] is ***true*** if and only if there is a *unique* shortest path from s to u. By definition usp[s] is ***true***. Be sure to prove both the correctness and time bound of your algorithm.

Answer:

```
Dijkstra (G,s):
  for each v ∈ V, do:
    d[v] ← ∞
    usp[v] ← false
  d[s] ← 0; usp[s] ← true
  s ← φ

  Add all vertices to Priority Q on key value d[v]

  while not EMPTY(Q), do:
    u ← EXTRACT_MIN(Q)
    Add u to s
    for each v in ADJ[u], do:
      if (d[u] + w(u,v) < d[v]), then:
        d[v] ← d[u] + w(u,v)
        usp[v] ← usp[u]
      else if (d[u] + w(u,v) = d[v]), then:
        usp[v] ← false
```

Since only an extra condition is added, the above algorithm runs as fast as Dijkstra's algorithm.

Hence, total time is:
$\Theta((|V| + |E|)\log|V|)$ for binary heaps
$\Theta(|V|\log|V| + |E|))$ for Fibnoacci heaps

Proof of correctness:
We know that in Dijkstra's SSSP algorithm when U is added to S, $d[u] = \delta(S, u)$ i.e. the minimum path weight from S to u.

During the course of the above algorithm, when a vertex in set S appears in the adjacency list of u, we compare its value $d[v]$ (shortest at that moment) to $d[u] + w(u, v)$.
If $d[u]+w(u, v) < d[v]$, then the $usp[v]$ is changed to $usp[u]$. i.e. if the shortest path to the predecessor node, then the shortest path to the current node is also not unique.
If they happen to be equal, then there exist multiple shortest path, and hence we update the $usp[v]$ to $false$.