Hw #5

4.5:

Initialize variables, setting dist[u] to 0, count[u] to 1, dist[v] to ∞, and count[w] to 0 for all w≠v

Then, can do BFS on Q, starting w/ u

Then, for each vertex x dequed from Q:

for each neighbor y of x:

If dist[y] = ∞, then dist[y] = dist[x]+1, and count[y] = count[x]. Then enqueue y

But if dist[y] = dist[x]+1, then count[y] += count[x]

The time complexity is O(|V|+|E|) when returning count[v], which is linear.

4.19:

Can modify Dijkstra's for vertex costs

Initialize cost[s] = c_s, cost[v] = ∞ for all v≠s, then priority queue H w/ s, c_s

while H isn't empty, extract vertex u with min cost[u]. For each edge u,v in the elements of E:

new_cost = cost[u] + l_u+v

If new_cost < cost[v], then cost[v] = new_cost, then update/insert v in H w/ priority cost[v].

finally, return cost array.

Time is O((|V|+|E|) log|V|)