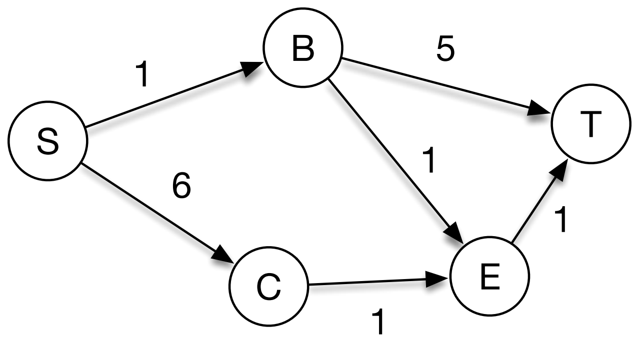
Name & UW NetID:

1. Run Dijkstra's shortest path algorithm in the following graph with vertex S as the source, and fill the table below with the results.



Vertex	Distance	Predecessor	Processed

2. Following is the pseudocode for Dijkstra's shortest path algorithm with binary heap. Do runtime analysis of this code. (Annotate the pseudocode with big-$\mathcal{O}$ values next to relevant statements/loops in the code.)

```
function Dijkstra(Graph G, Vertex source)                                ▷ with MPQ
    initialize distances to ∞
    source.dist = 0
    mark all vertices unprocessed
    initialize MPQ as a min priority queue
    add source with priority 0
    while MPQ is not empty do
        u = MPQ.getMin()
        for each edge (u,v) leaving u do
            if u.dist + w(u,v) < v.dist then
                if v.dist == ∞ then
                    MPQ.insert(v, u.dist + w(u, v))
                else
                    MPQ.decreasePriority(v, u.dist + w(u,v))
                end if
                v.dist = u.dist + w(u,v)
                v.predecessor = u
            end if
        end for
        mark u as processed
    end while
end function
```