Homework 7

Wednesday, May 4, 2022 12:42 AM

1. Single Source Shortest Path

Since we have an array to keep track of the distance, we can traverse this array at the end, find the length of the shortest path, and use a counter to count the number of paths with the same length as the shortest path.

```
Bellman-Ford (G, s)
d[v] = ∞ ; d[s] = 0 ; π[v] = null
repeat V-1 times:
    relax each edge, update d[v]
relax each edge one more time
    if any relaxation is constructive:
        output "negative-weight cycle detected"
for v in d:
    if d[v] ≤ min:
        min = d[v]
for v in d:
    if d[v] == min
        count = count + 1
return count
```

Time complexity : $O(V \cdot E)$

2. All Paths from Source to Destination

For a graph with V vertices: start and destination vertices are included in the path. The remaining $V-2$ vertices are either in the path or not $\Rightarrow O(2^{V-2})$

The remaining $V-2$ vertices are either in the path or not $\Rightarrow O(2^{V-2})$

Since in this case, we are dealing with a Directed Acyclic Graph (DAG), we can simply use Depth First Search to count all different paths from the source to the destination.

Pseudo code:

```
DFS ( s , d )
    count = 0 ;
    if ( s == d)
        count = count + 1 ;
    else for each neighbor i of s :
        count = count + DFS ( i , t ) ;
    return count ;
```

Time complexity : $O(V+E)$