University of Washington
Department of Computer Science and Engineering
CSE 417, Winter 2020
Yiliang Wang

February 5, 2020

Homework 4 Problem 3

Problem 3 (10 points):

Let $G = (V, E)$ be a directed graph with integral edge costs in $\{1, 2\}$. Give an $O(n + m)$ time algorithm that given vertices $s, t \in V$ finds a shortest path from s to t.

Answer:

algorithm:

```
def find_integral_shortest_path:
    # iterate through all edge and split length 2 edge
    n = len(g)
    for e in edges:
        if weight[e] == 2:
            g[e[0]].append(n+1)
            g[n + 1].append(e[1])
            n += 1

    visited = [0 for i in range(n)]
    queue = list()
    counter = 0
    queue.append(s)
    level = 0
    # use queue to construct a bfs, iterate through the graph,
    # and keep level number of each layer
    while len(queue)!= 0:
        for i in counter:
            cur = queue.pop(0)
            if cur == t:
                return level
            visited[counter] += 1
            for neighbor in g[cur]:
                if visited[neighbor] == 0:
                    queue.append(neighbor)
        count = len(queue)
        level += 1
    return inf
```

proof of correctness:
Spliting two-cost edge into 2 one-cost edges garantees the path len does not change. And it reduce all edge cost to 1, thus change this graph to a unweighted cost. Using Breadth first search can calculate the single source shortest path of unweighted path. Hence this algorithm can get return shortest path from s to t.

The time complexity is $O(n+m)$, the same as breadth first search:
- Spliting cost-two edge will traverse all edges, thus is $O(m)$;
- Adding new vertices is bound to $O(m)$, since there at most m cost-two edges. Spliting each cost-two edge will produce one new vertex and one new edge;
- After spliting, the number of vertices is at most $O(m)$, since the new number of edge is bound to 2m, and the new number of vertices is bound to 2n, therefore the time complexity for BFS of splitted graph is $O(2n+2m) = O(n+m)$
- $O(n+m) + O(m) + O(m) = O(n+m)$