BUSE AK 25469

CS301 HOMEWORK 3

1)a)

Given that G=(V,E)

Where s is the source node in our case İstanbul-Harem Bus station.

δk(s, s)=0

v is an element of V

Sub-problems : δ(s, v1), δ(s, v2), …,δ(s, vn) so the number of sub-problems is number of vertices in our case it is 81 since we are trying to find the shortest path to each city of Turkey from İstanbul-Harem bus station.

Any sub-path of the optimal path is optimal which tries to minimize the time, so it has an optimal substructure.

If we assume that there are no cycles than the recursive formulation has the property of topological ordering since it starts from the source node and goes on.

If we do not assume that there are no cycles, topological ordering is provided by

δ0(s, v)=infinity for s!=v

δk(s, s)=0 for k=0,1,…,Number of vertices -1 since the maximum path length could be number of vertices -1

δk(s, v)=min{ δ(k-1)(s, u)+w(u,v)}   where (u,v) is an element of Edge set.

W(u,v) = min((u,v)bus_time,(u,v)train_time+transfer_bw_bus_train) if we come to u with bus.

W(u,v) = min((u,v)train_time,(u,v)bus_time+transfer_bw_bus_train) if we come to u with train.

b)

G=(V,E)

δ(s, s)=0

δ(s, v)=INFINITY for every vertex

```
distance_list=[]
Func(src,v):
        If src is v then return 0
    If (u,v) is element of E
        If u.bus is True then:
                If (u,v)bus_time<(u,v)train_time+u_transfer_bw_bus_train)
                        δ(s, v)= Func(s, u)+(u,v)bus
        .
        . It is checked if bus and train used and the action is taken according to find the minimum time.
        .
return δ(s, v)
```

Number of vertices is n.

The time and space complexity analysis: The function recurs for each neighbor vertex u of vertex v as our target. So we check each vertex which is O(|V|) and for each vertex we check its adjacent vertex which gives us time complexity O(|V|^2).For every vertex we find the shortest path so for each vertex we need to store that data in a data structure such as vectors so that shows us that we need space as the number of vertices . That's why space complexity is O(|V|).

c)Bellman Ford algorithm is modified to solve our problem. Instead of w(u, v) part, a formulation is created to solve how to select bus or train and the cost of travelling with them.

```
d[s] ← 0
 for each v ∈ V – {s}
         do d[v] ← ∞ initialization
        for i ← 1 to |V| – 1
                do for each edge (u, v) ∈ E
                        do if d[v] > d[u] + w(u, v) then d[v] ← d[u] + w(u, v)
          for each edge (u, v) ∈ E
```

do if d[v] > d[u] + w(u, v) then report that a negative-weight cycle

d[v] = δ(s, v), if no negative-weight cycles. Time = O(|V||E|).

2)c) If we use the algorithm in part 1)c) with different vertex numbers and edges we obtain these results as time unit:

5=2.7000000000013125e-06

6=2.3999999999996247e-06

7=2.6999999999943736e-06

8=2.9000000000001247e-06