**ECE374 Assignment 7**

Due 04/03/2023

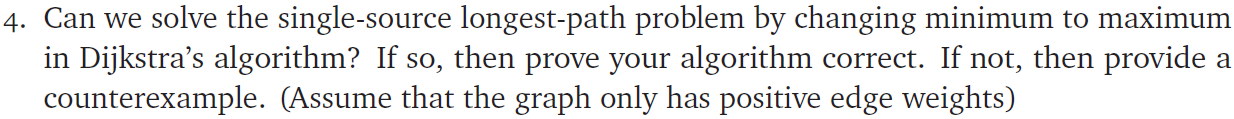
**Group & netid**

*Chen Si chensi3*

*Jie Wang jiew5*

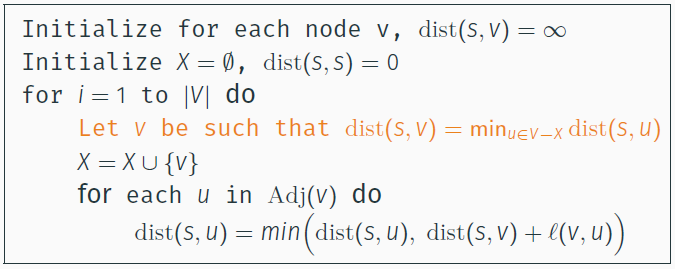
*Shitian Yang sy39*

**Problem 4**

****

Solution:

Dijkstra’s algorithm cited from lecture 17(p107):



After modification, the proposed algorithm would be –

**LongestPathDij**(G(V,E), s, t):

Initialize for each node v, dist(s,v)= –∞

X = ∅

dist(s,s) = 0

for i 🡨 1 to |V|:

Let v be such that dist(s, v) = max({dist(s,u)|u∈V\X})

X = X ∪ {v}

for u ∈ {u|u∈V\X and u∈Adj(v)}:

dist(s,u) = max(dist(s,u), dist(s,v)+l(v,u))

return dist(s,t)

A counterexample would be for the graph below, we have



0. initialize:

X={}

V\X = {A, B, C, D}

|  |  |
| --- | --- |
| dist(A, A) | 0 |
| dist(A, B) | -∞ |
| dist(A, C) | -∞ |
| dist(A, D) | -∞ |

1. i = 1

V\X = {A, B, C, D}, v = A

X=X∪{A}={A}

Update B, D

|  |  |
| --- | --- |
| dist(A, A) | 0 |
| dist(A, B) | max(-∞, 0+5) = 5 |
| dist(A, C) | -∞ |
| dist(A, D) | max(-∞, 0+2) = 2 |

2. i = 2

V\X = { B, C, D}, v = B

X=X∪{B}={A, B}

Update C

|  |  |
| --- | --- |
| dist(A, A) | 0 |
| dist(A, B) | 5 |
| dist(A, C) | max(-∞, 5+5) = 10 |
| dist(A, D) | 2 |

3. i = 3

V\X = {C, D}, v = C

X=X∪{C}={A, B, C}

|  |  |
| --- | --- |
| dist(A, A) | 0 |
| dist(A, B) | 5 |
| dist(A, C) | 10 |
| dist(A, D) | 2 |

4. i = 4

V\X = {D}, v = D

X=X∪{D}={A, B, C, D}

|  |  |
| --- | --- |
| dist(A, A) | 0 |
| dist(A, B) | 5 |
| dist(A, C) | 10 |
| dist(A, D) | 2 |

Note: can’t update C to 11 with AD=2 and CD=9, as C is not in the set V\X ∩ Adj(D).

5. Final output: dist(A, C) = 10

A significant problem in this counterexample is that when we first depart from A, we choose to follow A-B-C as AB is longer than AD. But after we update dist(A,C) along the path A-B-C and finally track down A-D, we couldn’t update C anymore though with a fresh, longer path from D, because C has already been marked as “Visited”. In a similar way, when a graph has its first edge off the start node in longest path not the longest among all edges departing the start node, we couldn’t update the final value at the terminal point as it has already been visited and thus is not in the set to update.

In this case, the maximum length this algorithm offers is 10 (A-B-C), but the actual longest path is 11 (A-D-C). Therefore, it’s problematic to use this modified Dijkstra algorithm to compute longest path.