DSA HW4
Yuhang Zhou
yz853
4/20/2019

Q1.

This graph is cyclic.

In the program, I create singly linked list as the data structure to store the graph data.

To check if this graph is cyclic, I use DFS, details in the source code.

Q2.

Prim's Algorithm: run time=3ms
Kruskal's Algorithm: run time=14ms
MST weight=10.46

The run time of Prim's algorithm is O(ElogV), while the run time of Kruskal's algorithm should be O(ElogE)

In this case, the dataset contains 250 vertices and 1273 edges. So it makes sense that the Kruskal's algorithm will take more time than the Prim's algorithm.

Q3.

Firstly, I'm using topological sort for this digraph.

The sorted result is: 5,1,3,6,4,7,0,2

The vertex 5 is the source, the distance from vertex n to 5 will be shown as Dist[n]

Shortest distance:

1. Start from 5
   5->5=0
   5->1=0.32
   5->4=0.35
   5->7=0.28

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | | 0.32 | | | 0.35 | 0 | | 0.28 |

2. From 1
   1->3=0.29,  5->3=0.32+0.29=0.61

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | | 0.32 | | 0.61 | 0.35 | 0 | | 0.28 |

3. From 3
   3->6=0.52,  5->6=0.61+0.52=1.13
   3->7=0.39,  5->7=0.61+0.39=1 > 0.28, longer, cannot replace

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | | 0.32 | | 0.61 | 0.35 | 0 | 1.13 | 0.28 |

4. From 6
   6->2=0.4,  5->2=0.4+1.13=1.53
   6->0=0.58,  5->0=0.58+1.13=1.71

6->4=0.93,  5->4=0.93+1.13=2.06 > 0.35, longer, can’t replace 0.35

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 1.71 | 0.32 | 1.53 | 0.61 | 0.35 | 0 | 1.13 | 0.28 |

5. From 4

4->0=0.38,  5->0=0.35+0.38=0.73 < 1.71, shorter, replace 1.71

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0.73 | 0.32 | 1.53 | 0.61 | 0.35 | 0 | 1.13 | 0.28 |

6. From 7

7->2=0.34,  5->2=0.34+0.28=0.62 < 1.53, shorter, replace 1.53

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0.73 | 0.32 | 0.62 | 0.61 | 0.35 | 0 | 1.13 | 0.28 |

7. From 0

0->2=0.26,  5->2=0.26+0.73=1 > 0.62, longer, cannot replace 0.62

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0.73 | 0.32 | 0.62 | 0.61 | 0.35 | 0 | 1.13 | 0.28 |

2 cannot go to any other vertex, done.

Shortest path:

0: 5->4->0

1: 5->1

2: 5->7->2

3: 5->1->3

4: 5->4

5: Source

6: 5->1->3->6

7: 5->7

Longest distance:

1. Start from 5

5->5=0

5->1=0.32

5->4=0.35

5->7=0.28

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | | 0.32 | | | 0.35 | 0 | | 0.28 |

2. From 1

1->3=0.29,  5->3=0.32+0.29=0.61

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | | 0.32 | | 0.61 | 0.35 | 0 | | 0.28 |

3. From 3

3->6=0.52,  5->6=0.61+0.52=1.13

3->7=0.39,  5->7=0.61+0.39=1 > 0.28, longer, replace 0.28

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | | 0.32 | | 0.61 | 0.35 | 0 | 1.13 | 1 |

4. From 6

6->2=0.4,　5->2=0.4+1.13=1.53
6->0=0.58,　5->0=0.58+1.13=1.71
6->4=0.93,　5->4=0.93+1.13=2.06 > 0.35, longer, replace 0.35

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 1.71 | 0.32 | 1.53 | 0.61 | 2.06 | 0 | 1.13 | 1 |

5. From 4
   4->0=0.38,　5->0=0.38+2.06=2.44 > 1.71, longer, replace 1.71
   4->7=0.37,　5->7=0.37+2.06=2.43 > 1, longer replace 1

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 2.44 | 0.32 | 1.53 | 0.61 | 2.06 | 0 | 1.13 | 2.43 |

6. From 7
   7->2=0.34,　5->2=0.34+2.43=2.77 > 1.53, longer, replace 1.53

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 2.44 | 0.32 | 2.77 | 0.61 | 2.06 | 0 | 1.13 | 2.43 |

7. From 0
   0->2=0.26,　5->2=0.26+2.44=2.7 < 2.77, shorter, cannot replace 2.77

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 2.44 | 0.32 | 2.77 | 0.61 | 2.06 | 0 | 1.13 | 2.43 |

2 cannot go to any other vertex, done.

Longest path:
0: 5->1->3->6->4->0
1: 5->1
2: 5->1->3->6->4->7->2
3: 5->1->3
4: 5->1->3->6->4
5: Source
6: 5->1->3->6
7: 5->1->3->6->4->7

Q4.a
Set vertex 0 as the source
1. 0->4=0.38
   0->2=0.26

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0 | | 0.26 | | 0.38 | | | |

2. 2->7=0.34,　0->7=0.34+0.26=0.6
   4->7=0.37,　0->7=0.37+0.38=0.75>0.6
   4->5=0.35,　0->5=0.35+0.38=0.73

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0 | | 0.26 | | 0.38 | 0.73 | | 0.6 |

3. 5->4=0.35,　0->4=0.35+0.73>0.38
   5->7=0.28,　0->7=0.28+0.73>0.6
   5->1=0.32,　0->1=0.32+0.73=1.05
   7->5=0.28,　0->5=0.28+0.6>0.73

7->3=0.39, 0->3=0.39+0.6=0.99

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0 | 1.05 | 0.26 | 0.99 | 0.38 | 0.73 | | 0.6 |

4. 1->3=0.29, 0->3=0.29+1.05>0.99
   3->6=0.52, 0->6=0.52+0.99=1.51

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0 | 1.05 | 0.26 | 0.99 | 0.38 | 0.73 | 1.51 | 0.6 |

5. 6->2=-1.2, 0->2=0.51-1.2<0
   6->0=-1.4, 0->0=1.51-1.4=0.11>0
   6->4=-1.25, 0->4=1.51-1.25=0.26<0.38

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0 | 1.05 | 0.26 | 0.99 | 0.26 | 0.73 | 1.51 | 0.6 |

6. 4->5=0.35, 0->5=0.26+0.35=0.61<0.73
   4->7=0.37, 0->7=0.37+0.35>0.6

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0 | 1.05 | 0.26 | 0.99 | 0.26 | 0.61 | 1.51 | 0.6 |

7. 5->4=0.35, 0->4=0.35+0.61 >0.38
   5->7=0.28, 0->7=0.28+0.61 >0.6
   5->1=0.32, 0->1=0.32+0.61 =0.93<1.05

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0 | 0.93 | 0.26 | 0.99 | 0.26 | 0.61 | 1.51 | 0.6 |

8. 1->3=0.29, 0->3=0.93+0.29>0.99

Done

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0 | 0.93 | 0.26 | 0.99 | 0.26 | 0.61 | 1.51 | 0.6 |

Q4.b

Set vertex 0 as the source

1. 0->4=0.38
   0->2=0.26

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0 | | 0.26 | | 0.38 | | | |

2. 2->7=0.34, 0->7=0.34+0.26=0.6
   4->7=0.37, 0->7=0.37+0.38=0.75>0.6
   4->5=0.35, 0->5=0.35+0.38=0.73

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0 | | 0.26 | | 0.38 | 0.73 | | 0.6 |

3. 5->4=-0.66, 0->4=0.73-0.66=0.07
   5->7=0.28, 0->7=0.28+0.73=1.01>0.6
   5->1=0.32, 0->1=0.32+0.73=1.05
   7->5=0.28, 0->5=0.28+0.6>0.73
   7->3=0.39, 0->3=0.39+0.6=0.99

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0 | 1.05 | 0.26 | 0.99 | 0.07 | 0.73 | | 0.6 |

4. 1->3=0.29, 0->3=0.29+1.05>0.99
   3->6=0.52, 0->6=0.52+0.26=0.78
   4->5=0.35, 0->5=0.35+0.07=0.42<0.73
   4->7=0.37, 0->7=0.37+0.07=0.44<0.6

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Dist[ ] | 0 | 1.05 | 0.26 | 0.99 | 0.07 | 0.42 | 0.78 | 0.44 |

5. 5->4=-0.66, 0->4=0.07-0.66=-0.59<0.07

In step 4, vertex 4 leads to dist[5] decreasing, then in step 5, vertex 5 leads to dist[4] decreasing again like that in the step 3. So the program comes to a negative cycle, the Bellman-Ford algorithm will stuck.

Q5.

For the DFS, I use a recursive function. For the BFS, I use a queue to record those vertices which already been visited, while its surround vertices not yet.
Details in the source code.

Q6.

For the dataset of Q4.a, the result is

```
dist to vertex 0 : 0
dist to vertex 1 : 0.93
dist to vertex 2 : 0.26
dist to vertex 3 : 0.99
dist to vertex 4 : 0.26
dist to vertex 5 : 0.61
dist to vertex 6 : 1.51
dist to vertex 7 : 0.6
```

It's the same as what I get in the Q4.a.
For the dataset of Q4.b, the problem is stuck.
In the while loop, I use an priority queue to save vertices which will be processed in the next loop. So I add an output in the while loop to see which vertex is popped, and which one is pushed. Output:

```
pop vertex 5, dist = -1.44
push vertex 4, dist = -2.1
push vertex 1, dist = -1.12
pop vertex 4, dist = -2.1
push vertex 5, dist = -1.75
push vertex 7, dist = -1.73
```

So I found that this 6 progress keep repeating, and the distance keeps decreasing, that means this program comes to a negative cycle.