CS 311 HW7 Graph Algorithms Part 1
MST and Shortest Path (based on week 11 - 12)
==========

DUE: Week 14 Monday

TOTAL 32 points Your score:

*NAME: Eduardo Martinez

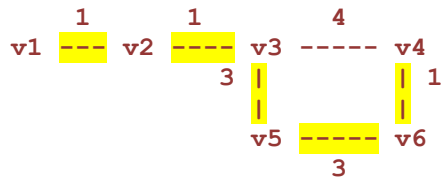
*DATE SUBMITTED: 11/28/16

Purpose: To be able to follow graph algorithms

Review Questions from Week 12 and 13:

1. Using the following graph, do the Prim/Dijkstra algorithm for finding a MST [1 per prompt=12pts] Your score:

Start with v1 and complete the following.
Give the grand total cost at the bottom.



Tree: v1
Fringe: v2
Select v1-v2 with cost 1

Tree: v1 v2
Fringe: v3
Select: v2-v3 with cost 1

Tree: v1 v2 v3
Fringe: v4 and v5
Select: v3-v5 with cost 3

Tree: v1 v2 v3 v5
Fringe: v4 and v6
Select: v5-v6 with cost 3

Tree: v1 v2 v3 v5 v6
Fringe: v4
Select: v6-v4 with cost 1

Highlight the MST edges in the above drawing.

TOTAL COST OF MST IS: 1+1+3+3+1=9

2. SHORTEST PATH [20pts] Your score:

Use the following directed graph.

A to B is 4
A to F is 2

B to A is 1
B to C is 3
B to D is 4

C to A is 6
C to B is 3
C to D is 7

D to A is 6
D to E is 2

E to D is 5

F to D is 2
F to E is 3

a) Do Dijkstra's shortest path algorithm starting with C ending with E.

Trace the algorithm using **the same format** as I used in Notes-12A. (Remember the tables???) **Run my solution program to help you.**
Note that you may stop as soon as E is added to the Tree.

Initially:
Tree has: C
Fringe (F*) has: A, B and D
DistTo of these are initialized to be the edge weights from C.

Tree: C

	DistTo	CandidateEdge
F*	A 6	from C
F*	B 3	from C
F*	D 7	from C
	E inf	
	F inf	

Step1:
Pick B (show that B is T* in the table)
Tree (T*) now has: C, B
B is next to A, C, D. A and D are already in Fringe.
DistTo to all Fringe vertices when going through B:

- **DistTO[A] = DistTo[B] + weight(B,A) = 4 update (it is better)**
- **DistTO[D] = DistTo[B] + weight(B,D) = 7 no change (it is not better)**

Tree: C, B

		DistTo	CandidateEdge
F*	A	4	from B updated

T*	B	3	from C
F*	D	7	from C
	E	inf	
	F	inf	

Continue the trace using the same format until E is in the Tree.[16]

Step 2:

Pick A (show that A is T* in the table)

Tree (T*) now has:C,B,A

A is next to: B,F

DistTo to these Fringe (F*) vertices when going through A:

- DistTO[B] = DistTo[A] + weight(A,B) = 8 no change (it is not better)
- DistTO[F] = DistTo[A] + weight(A,F) = 6 update (it is better)

(indicate new or updated in the table)

Tree: C,B,A

		DistTo	CandidateEdge
T*	A	4	from B
T*	B	3	from C
F*	D	7	from C
	E		
F*	F	6	from A

Step 3: etc. (copy the above, paste and update).

Pick F (show that F is T* in the table)

Tree (T*) now has:C,B,A,F

F is next to: D,E

DistTo to these Fringe (F*) vertices when going through A:

- DistTO[D] = DistTo[F] + weight(F,D) = 7 update (it is better)
- DistTO[E] = DistTo[F] + weight(F,E) = 9 no change (it is not better)

(indicate new or updated in the table)

Tree: C,B,A,F

		DistTo	CandidateEdge
T*	A	4	from B
T*	B	3	from C
F*	D	7	from C
F*	E	9	from F
T*	F	6	from A

Step 4:

Pick D (show that D is T* in the table)

Tree (T*) now has:C,B,A,F,D

D is next to: A,E

DistTo to these Fringe (F*) vertices when going through A:

- DistTO[A] = DistTo[D] + weight(D,A) = 6 no change (it is not better)
- DistTO[E] = DistTo[D] + weight(D,E) = 9 update (it is better)

(indicate new or updated in the table)

Tree: C,B,A,F,D

		DistTo	CandidateEdge
T*	A	4	from B
T*	B	3	from C
T*	D	7	from C
F*	E	9	from F
T*	F	6	from A

STEP 5:

Pick E (show that E is T* in the table)

Tree (T*) now has:C,B,A,F,D,E

E is next to: D

DistTo to these Fringe (F*) vertices when going through A:

- **DistTO[D] = DistTo[E] + weight(E,D) = 14 no change**

(indicate new or updated in the table)

Tree: C,B,A,F,D,E

		DistTo	CandidateEdge
T*	A	4	from B
T*	B	3	from C
T*	D	7	from C
T*	E	9	from F
T*	F	6	from A

b) What was the path from C to E found by the algorithm? Indicate the vertices on the path.[2]

C-B-A-F-E

c) How did you retrieve that path from the table? (What information from the very last table did you use?)[2]

I started with e, in this case, and kept going back to 'from' or 'candidate edge' until it got to c or the root. Then the shortest path would be the reverse of that pathway i found from e to c because I wanted the shortest pathway from c to e

Submit this file:

- **This assignment sheet with your answers**