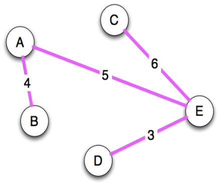
Programming Assignment 5 - MST

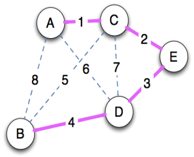
Test Cases

Graphs Used as Test Cases

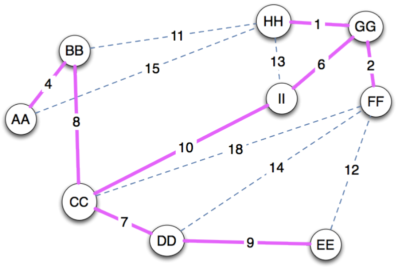
Graph1



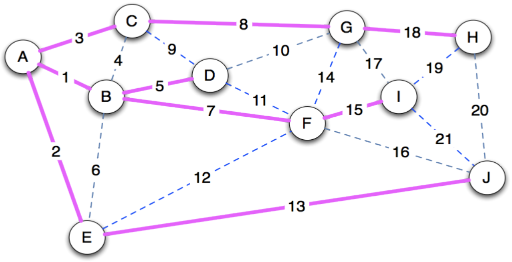
Graph2



Graph3



Graph4



NOTE:: In each of the tests below, the "Expected Ouput" shows the result in readable printed form, but the actual grader checked the resulting data structures in your program.

1. **Method: MST.initialize**

   Tested with 4 test cases. In each case partial trees are initialized from the constructed graphs.

Points	Test Case	Expected Output

5	graph1.txt	`Vertices: A PQ: (A B 4) (A E 5)`
		`Vertices: B PQ: (B A 4)`
		`Vertices: C PQ: (C E 6)`
		`Vertices: D PQ: (D E 3)`
		`Vertices: E PQ: (E D 3) (E C 6) (E A 5)`
		`-----------------------------------------------`

5	graph2.txt	`Vertices: A PQ: (A C 1) (A D 6) (A B 8)`
		`Vertices: B PQ: (B D 4) (B C 5) (B A 8)`
		`Vertices: C PQ: (C A 1) (C E 2) (C B 5) (C D 7)`
		`Vertices: D PQ: (D E 3) (D A 6) (D B 4) (D C 7)`
		`Vertices: E PQ: (E C 2) (E D 3)`
		`-----------------------------------------------`

5	graph3.txt	`Vertices: AA PQ: (AA BB 4) (AA HH 15)`
		`Vertices: BB PQ: (BB AA 4) (BB HH 11) (BB CC 8)`
		`Vertices: CC PQ: (CC DD 7) (CC BB 8) (CC II 10) (CC FF 18)`
		`Vertices: DD PQ: (DD CC 7) (DD FF 14) (DD EE 9)`
		`Vertices: EE PQ: (EE DD 9) (EE FF 12)`
		`Vertices: FF PQ: (FF GG 2) (FF EE 12) (FF DD 14) (FF CC 18)`
		`Vertices: GG PQ: (GG HH 1) (GG II 6) (GG FF 2)`
		`Vertices: HH PQ: (HH GG 1) (HH II 13) (HH BB 11) (HH AA 15)`
		`Vertices: II PQ: (II GG 6) (II HH 13) (II CC 10)`
		`-----------------------------------------------`

5	graph4.txt	Vertices: A PQ: (A B 1) (A C 3) (A E 2) Vertices: B PQ: (B A 1) (B C 4) (B E 6) (B F 7) (B D 5) Vertices: C PQ: (C A 3) (C B 4) (C G 8) (C D 9) Vertices: D PQ: (D B 5) (D C 9) (D G 10) (D F 11) Vertices: E PQ: (E A 2) (E B 6) (E F 12) (E J 13) Vertices: F PQ: (F B 7) (F E 12) (F D 11) (F J 16) (F G 14) (F I 15) Vertices: G PQ: (G C 8) (G D 10) (G I 17) (G H 18) (G F 14) Vertices: H PQ: (H G 18) (H J 20) (H I 19) Vertices: I PQ: (I F 15) (I G 17) (I H 19) (I J 21) Vertices: J PQ: (J E 13) (J F 16) (J H 20) (J I 21) ---

2. **Method: PartialTreeList.remove**

   This method was tested by initializing with graph1.txt (using our version of the MST.initialize method), then applying a sequence of removes on the resulting partial tree list. After each remove, the returned partial tree was checked.

Points	Test Case	Expected Output
2	After 1st remove	Vertices: A PQ: (A B 4) (A E 5)
2	After 2nd remove	Vertices: B PQ: (B A 4)
2	After 3rd remove	Vertices: C PQ: (C E 6)
2	After 4th remove	Vertices: D PQ: (D E 3)
2	After 5th remove	Vertices: E PQ: (E D 3) (E C 6) (E A 5)

3. **Method: PartialTreeList.removeTreeContaining**

   This method was tested as follows. A correct initial partial tree list was created by using our version of the MST.initialize method, once for graph1.txt, and then once for graph2.txt.

   For each of these, a remove was done, using our correct version of PartialTreeList.remove, then your removeTreeContaining method was run. The returned partial tree and the resulting partial tree list were then both checked for correctness.

Points	Test Case	Expected Output
7	Initialized with graph1.txt Removed partial tree A from partial tree list. Called method for C	Partial Tree - Vertices: B PQ: (B A 4) Partial Tree List: Vertices: C PQ: (C E 6) Vertices: D PQ: (D E 3) Vertices: E PQ: (E D 3) (E C 6) (E A 5) ---
6	Removed partial tree C from partial tree list Called method for E	Partial Tree - Vertices: E PQ: (E D 3) (E C 6) (E A 5) Partial Tree List: Vertices: D PQ: (D E 3) ---
7	Initialized with graph2.txt Removed partial tree A from partial tree list Called method for C	Partial Tree - Vertices: C PQ: (C A 1) (C E 2) (C B 5) (C D 7) Partial Tree List: Vertices: B PQ: (B D 4) (B C 5) (B A 8) Vertices: D PQ: (D E 3) (D A 6) (D B 4) (D C 7) Vertices: E PQ: (E C 2) (E D 3) -------------------- ---
	Removed partial tree B	Partial Tree - Vertices: D PQ: (D E 3) (D A 6) (D B 4) (D C 7)

5	from partial tree list Called method for D	Partial Tree List: Vertices: E PQ: (E C 2) (E D 3)

4. **Method: MST.execute**

Points	Tested Input	Expected Output
8	graph3.txt	(AA BB 4) (CC DD 7) (EE DD 9) (FF GG 2) (HH GG 1) (II GG 6) (BB CC 8) (II CC 10)
8	graph4.txt	(A B 1) (C A 3) (D B 5) (E A 2) (F B 7) (G C 8) (H G 18) (I F 15) (J E 13)
4	graph1.txt	(A B 4) (C E 6) (D E 3) (A E 5)