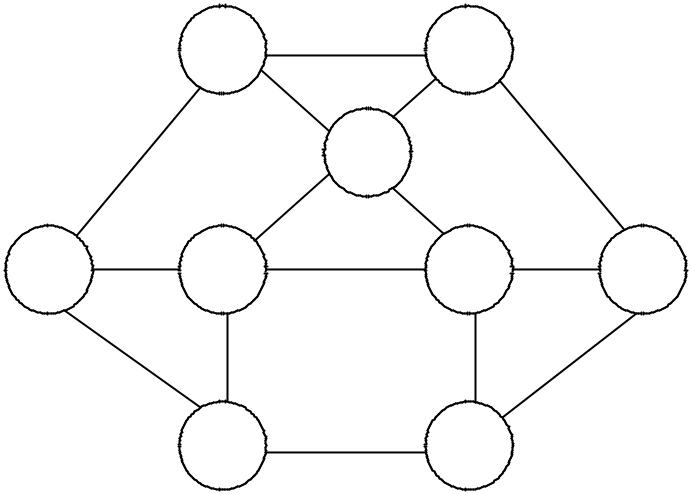
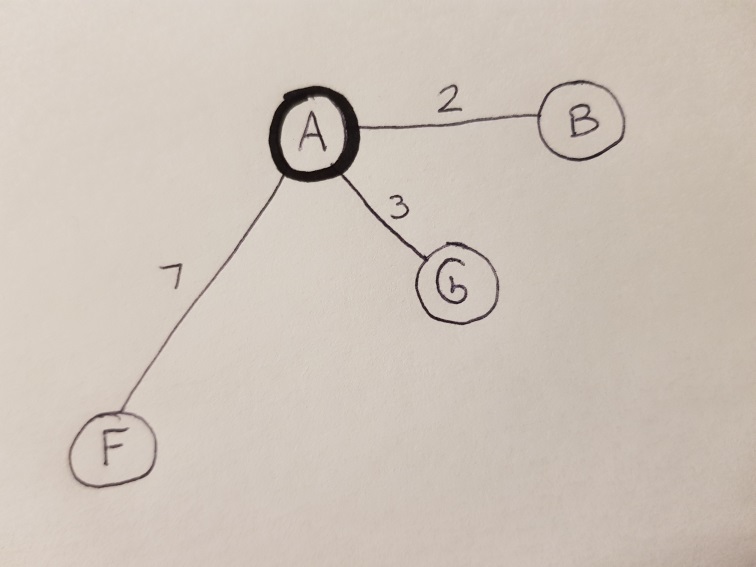
**CMSC 451 Homework 5**

1. Execute Prim’s minimum spanning tree algorithm by hand on the graph below showing how the data structures evolve specifically indicating when the distance from a fringe vertex to the tree is updated. Clearly indicate which edges become part of the minimum spanning tree and in which order. Start at vertex A.



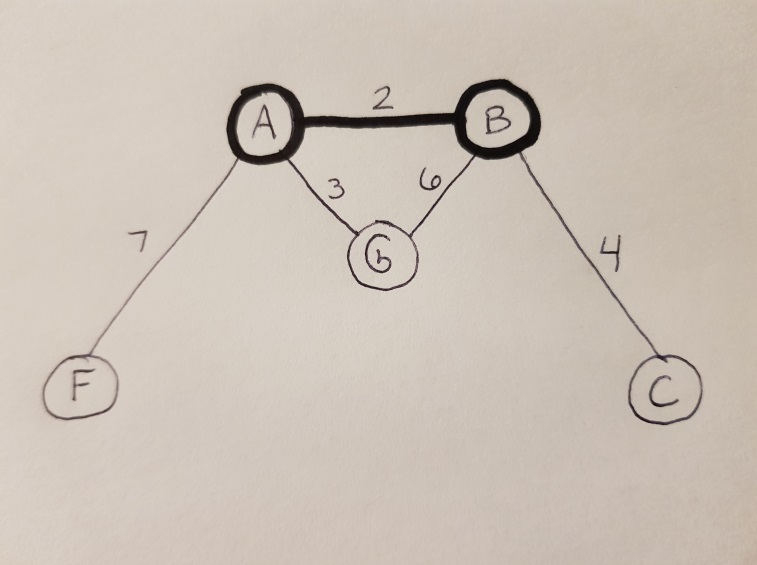
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | A |  | 2 | B |  |  |  |
|  |  |  | 3 |  |  |  |
|  |  |  |  | 6 |  |  |  |
|  | 7 |  |  |  |  | 4 |  |
|  |  |  |  | G |  |  |
|  |  |  | 1 |  | 3 |  |  |  |
|  |  |  |  |  |  |  |  |
| F | 5 | I |  | 4 | H | 2 | C |  |
|  |  |  |  |  |  |
|  | 6 |  | 2 |  |  | 8 | 2 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  | E |  | 1 | D |  |  |  |
|  |  |  |  |  |  |  |  |

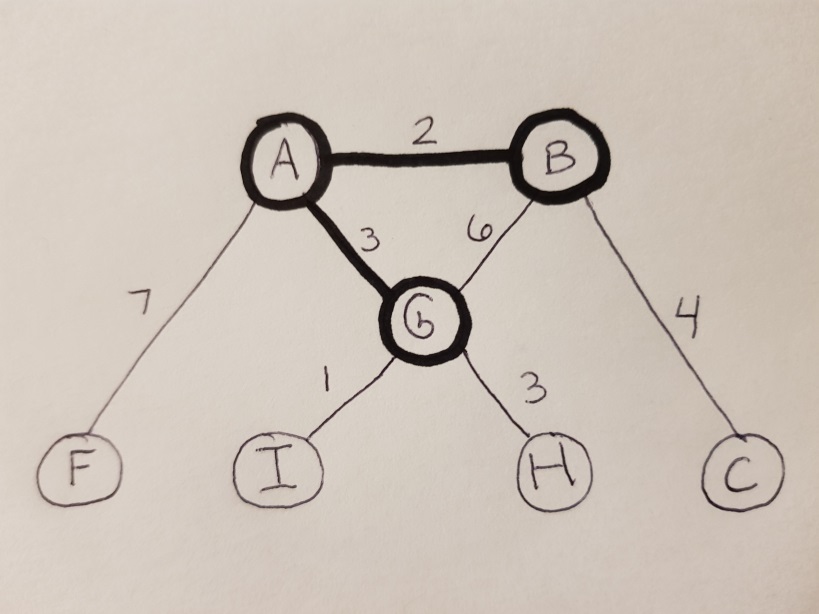
|  |  |  |
| --- | --- | --- |
| A | U | 0 minimum |
| B | U |  |
| C | U |  |
| D | U |  |
| E | U |  |
| F | U |  |
| G | U |  |
| H | U |  |
| I | U |  |



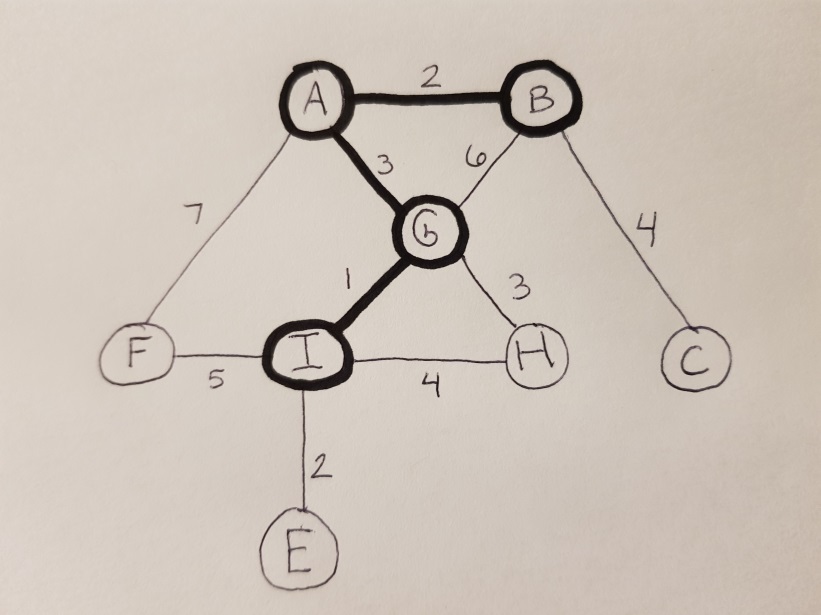
|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | F | 2 minimum |
| C | U |  |
| D | U |  |
| E | U |  |
| F | F | 7 |
| G | F | 3 |
| H | U |  |
| I | U |  |

|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | T |  |
| C | F | 4 |
| D | U |  |
| E | U |  |
| F | F | 7 |
| G | F | 3 minimum |
| H | U |  |
| I | U |  |



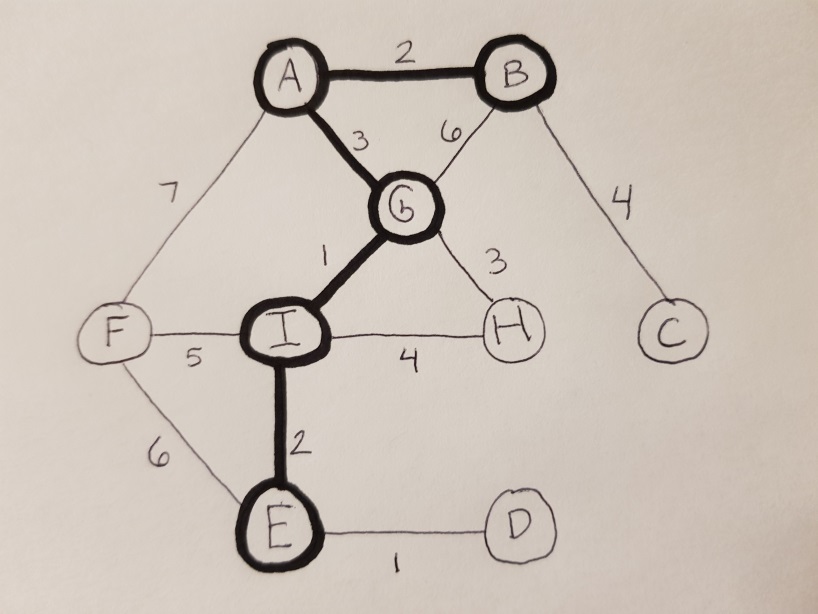


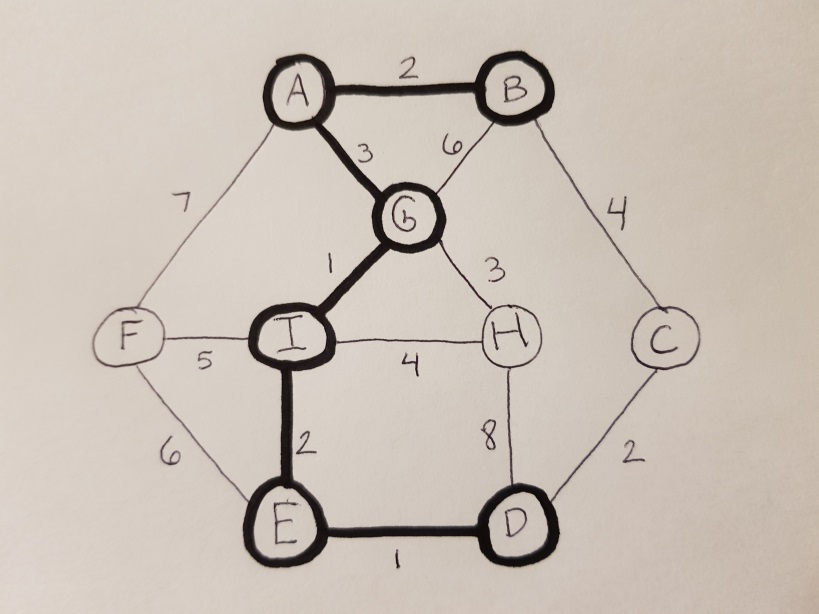
|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | T |  |
| C | F | 4 |
| D | U |  |
| E | U |  |
| F | F | 7 |
| G | T |  |
| H | F | 3 |
| I | F | 1 minimum |



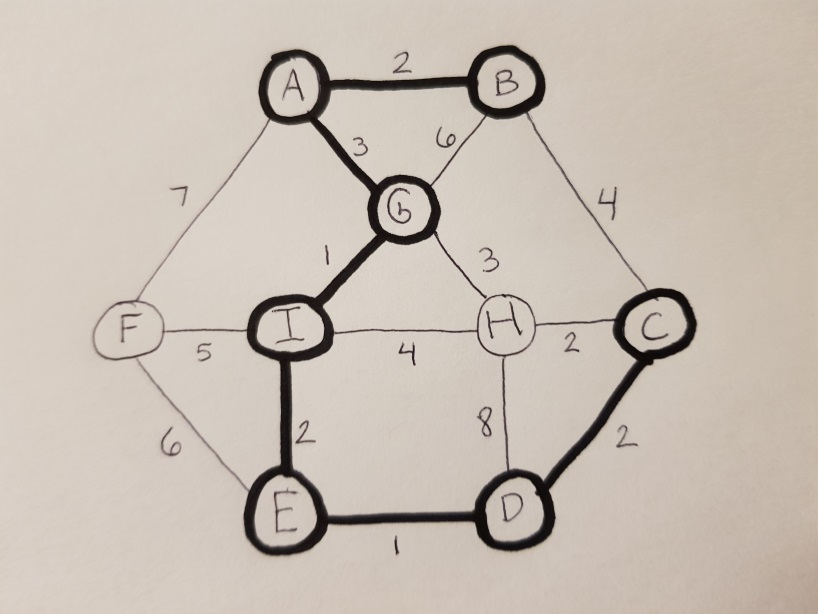
|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | T |  |
| C | F | 4 |
| D | U |  |
| E | F | 2 minimum |
| F | F | 5  updated |
| G | T |  |
| H | F | 3 |
| I | T |  |

|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | T |  |
| C | F | 4 |
| D | F | 1  minimum |
| E | T |  |
| F | F | 5 |
| G | T |  |
| H | F | 3 |
| I | T |  |

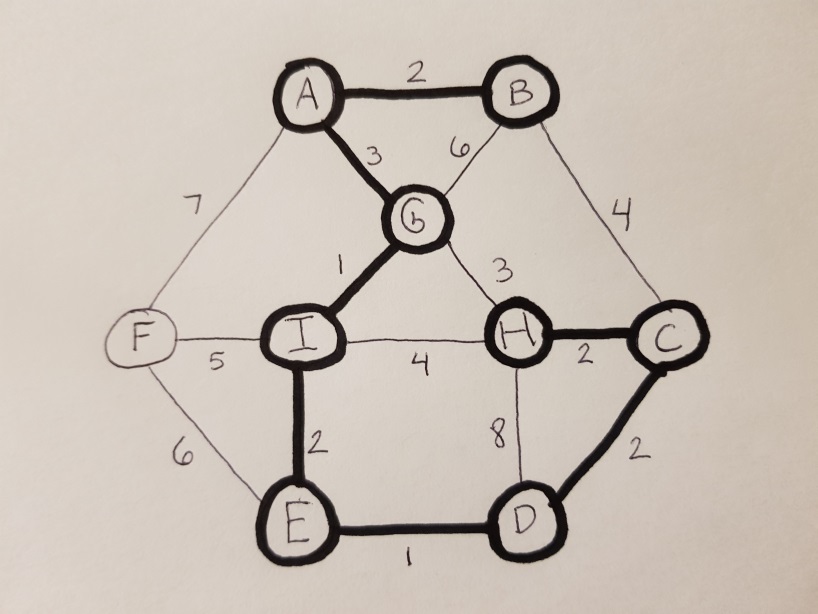




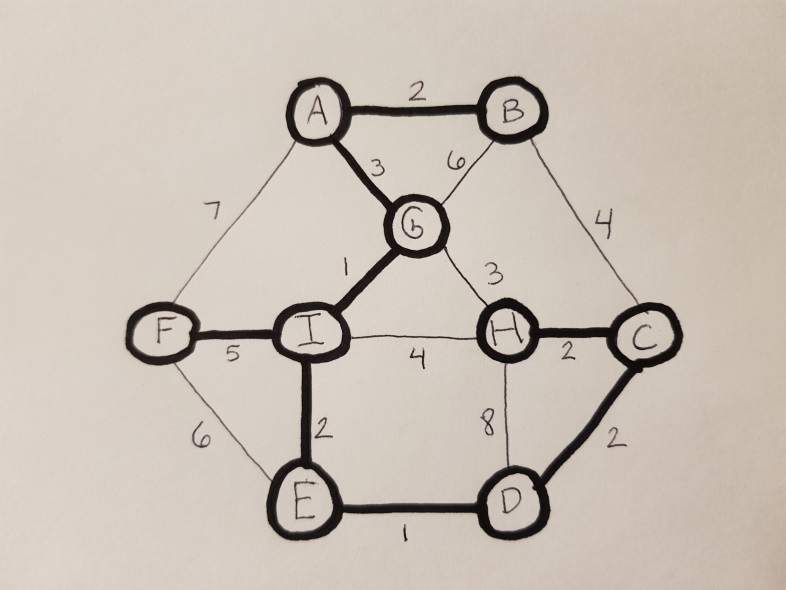
|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | T |  |
| C | F | 2 minumum, updated |
| D | T |  |
| E | T |  |
| F | F | 5 |
| G | T |  |
| H | F | 3 |
| I | T |  |



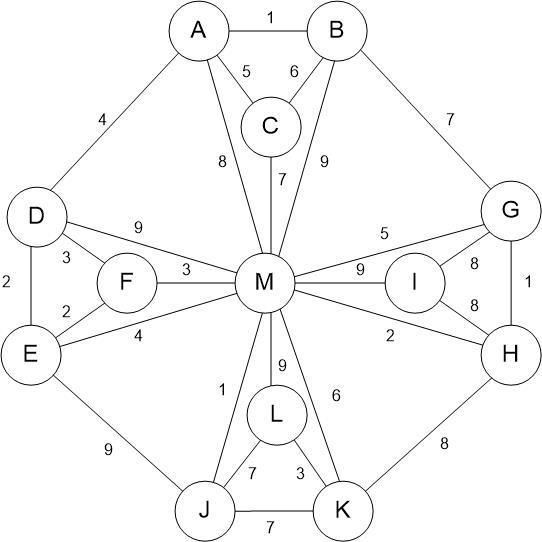
|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | T |  |
| C | T |  |
| D | T |  |
| E | T |  |
| F | F | 5 |
| G | T |  |
| H | F | 2 minimum, updated |
| I | T |  |



|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | T |  |
| C | T |  |
| D | T |  |
| E | T |  |
| F | F | 5 minimum |
| G | T |  |
| H | T |  |
| I | T |  |



|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | T |  |
| C | T |  |
| D | T |  |
| E | T |  |
| F | T |  |
| G | T |  |
| H | T |  |
| I | T |  |

1. Execute Kruskal’s algorithm on the weighted tree shown below. Assume that edges of equal weight will be in the priority queue in alphabetical order. Clearly show what happens each time an edge is removed from the priority queue and how the dynamic equivalence relation changes on each step and show the final minimum spanning tree that is generated.

Below is the action each time an edge is removed from the priority queue and the state of the dynamic equivalence relation after that edge has been processed. Only the equivalence classes that contain more than a single vertex are shown:

Choose AB 1 (A,B)

Choose GH 1 (A,B) (G,H)

Choose JM 1 (A,B) (G,H) (J,M)

Choose DE 2 (A,B) (G,H) (J,M) (D,E)

Choose EF 2 (A,B) (G,H) (J,M) (D,E,F)

Choose HM 2 (A,B) (G,H,J,M) (D,E,F)

Skip DF 3

Choose FM 3 (A,B) (D,E,F,G,H,J,M)

Choose KL 3 (A,B) (D,E,F,G,H,J,M) (K,L)

Choose AD 4 (A,B,D,E,F,G,H,J,M) (K,L)

Skip EM 4

Choose AC 5 (A,B,C,D,E,F,G,H,J,M) (K,L)

Skip GM 5

Skip BC 6

Choose KM 6 (A,B,C,D,E,F,G,H,J,K,L,M)

Skip BG 7

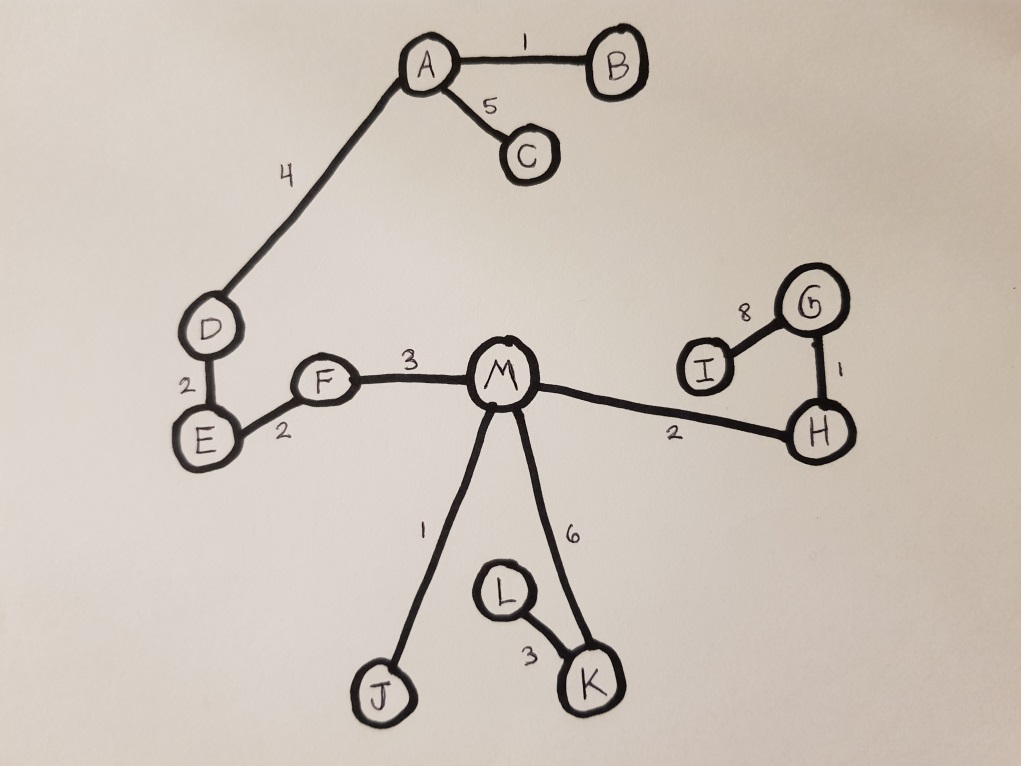
Skip CM 7

Skip JK 7

Skip JL 7

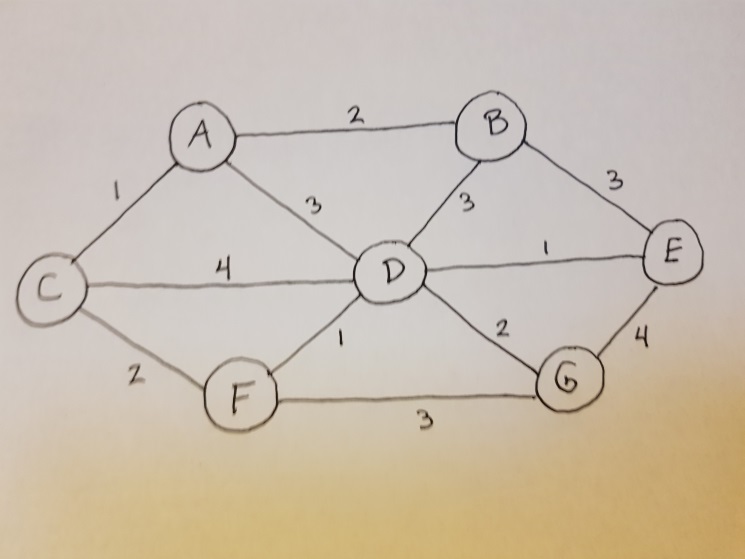
Skip AM 8

Choose GI 8 (A,B,C,D,E,F,G,H,I,J,K,L,M)



1. Give an example of a weighted graph for which the minimum spanning tree is unique. Indicate what the minimum spanning tree is for that graph. Give another example of a weighted graph that has more than one minimum spanning tree. Show two different minimum spanning trees for that graph. What determines whether a graph has more than one minimum spanning tree?

Graph A



Using Kruskal’s algorithm, we get the following unique MST for Graph A:

Choose AC 1 (A,C)

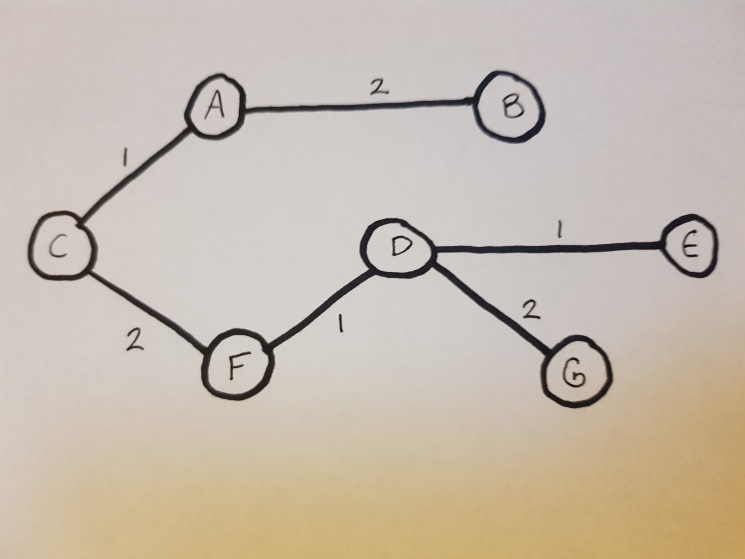
Choose DE 1 (A,C) (D,E)

Choose DF 1 (A,C) (D,E,F)

Choose AB 2 (A,B,C) (D,E,F)

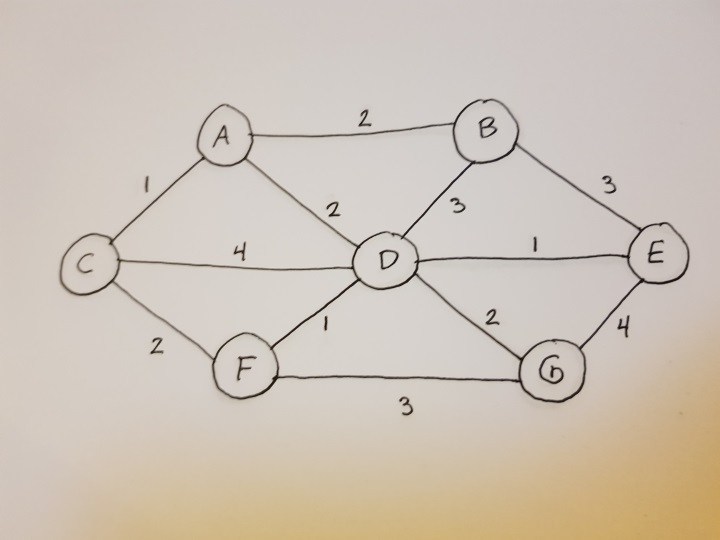
Choose CF 2 (A,B,C,D,E,F)

Choose DG 2 (A,B,C,D,E,F,G)



Graph w/ two minimum spanning trees:

Graph B



Graph B is almost identical to Graph A, except that the edge weight of AD was changed from 3 to 2. Using Kruskal’s algorithm, we get two possible MSTs:

Choose AC 1 (A,C)

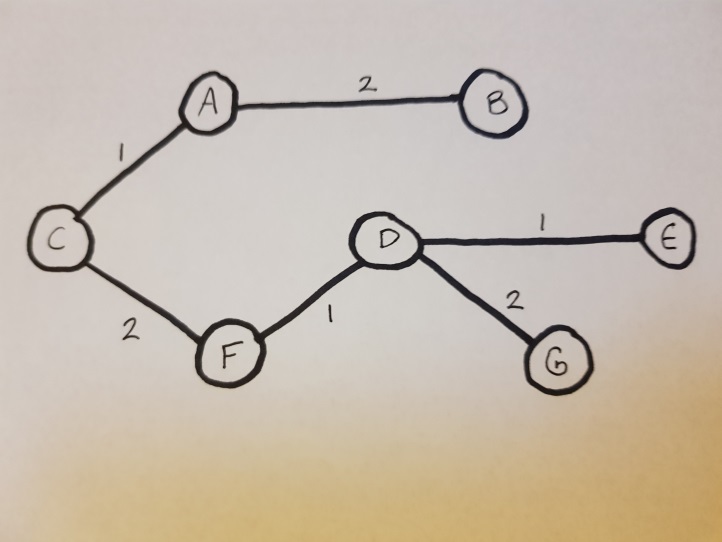
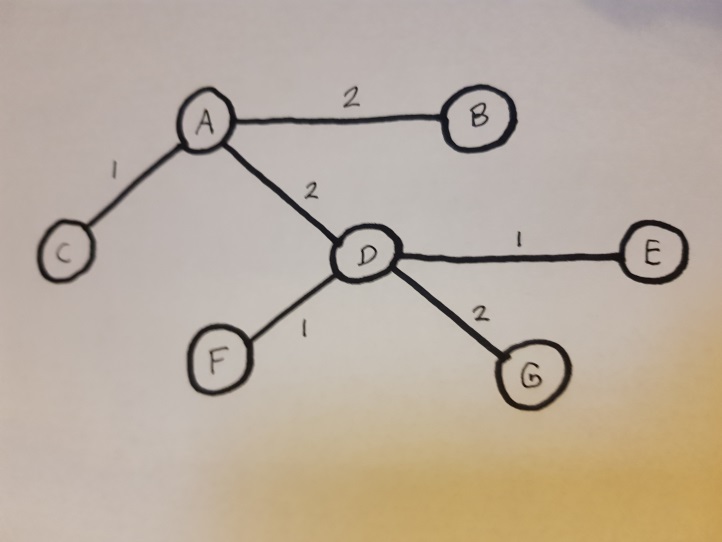
Choose DE 1 (A,C) (D,E)

Choose DF 1 (A,C) (D,E,F)

Choose AB 2 (A,B,C) (D,E,F)

**Choose CF 2 (A,B,C,D,E,F) -OR- Choose AD 2 (A,B,C,D,E,F)**

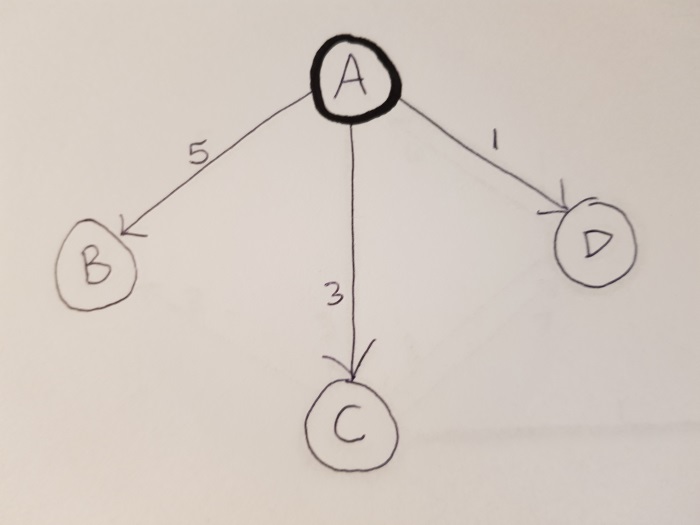
Choose DG 2 (A,B,C,D,E,F,G)



As we can see from graphs A and B, the deciding factor in whether a graph has more than one MST is the whether the subtrees of that graph can be connected by more than one edge with the same minimal edge width. In this case, the subtrees (A,B,C) and (D,E,F,G) can be connected by either edge CF or edge AD, both with a minimal edge width of 2.

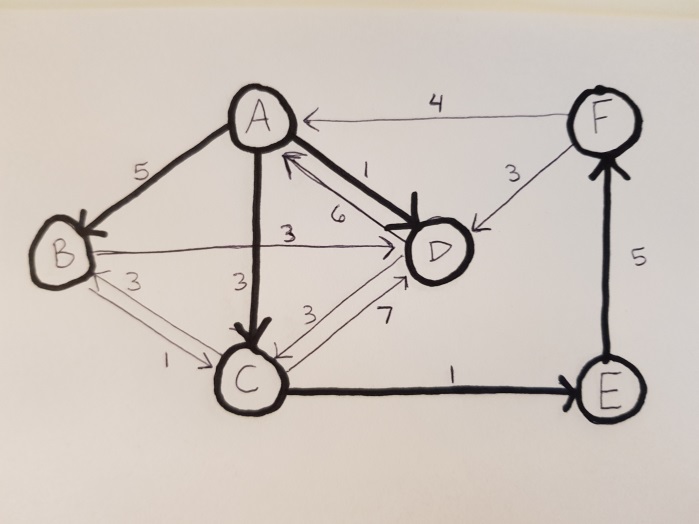
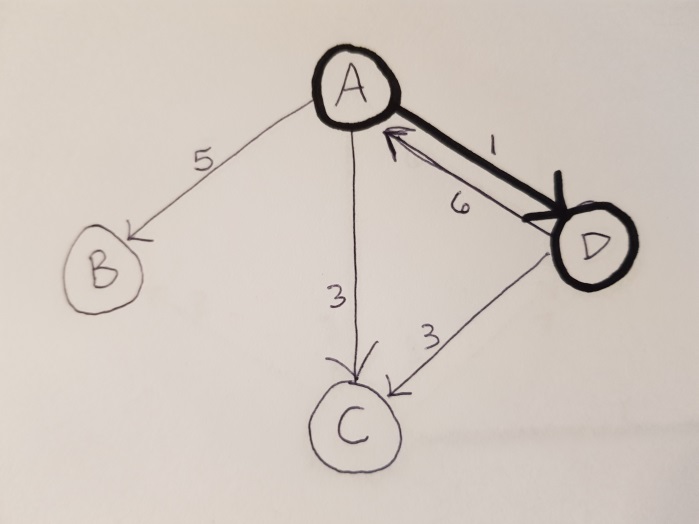
1. Given the following adjacency lists (with edge weights in parentheses) for a directed graph:
   1. B(5), C(3), D(1)
   2. C(1), D(3)
   3. B(3), D(7), E(1)
   4. A(6), C(3)
   5. F(5)
   6. D(3), A(4)

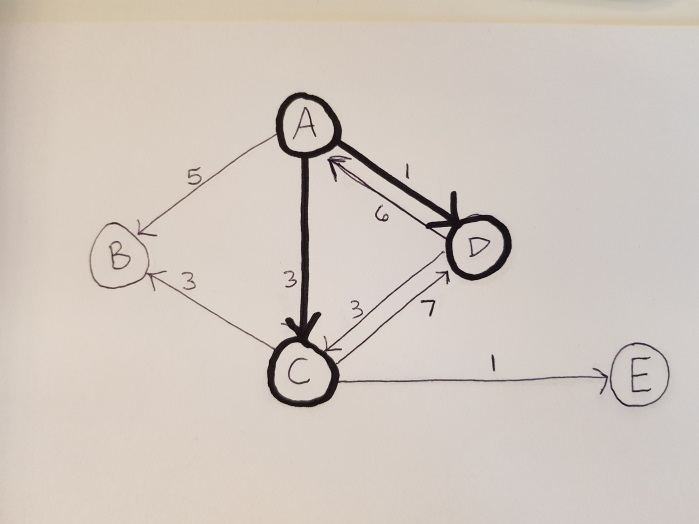
Execute Dijkstra’s shortest-path algorithm by hand on this graph, showing how the data structures evolve, with *A* as the starting vertex. Clearly indicate which edges become part of the shortest path and in which order.



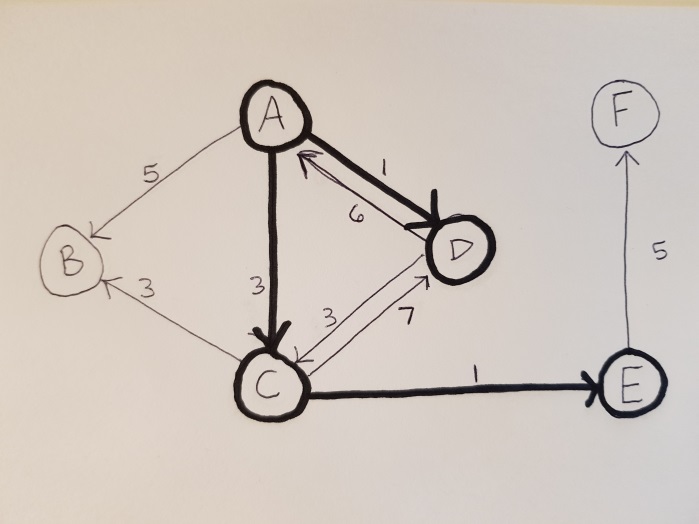
|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | F | 5 |
| C | F | 3 |
| D | F | 1 minimum |
| E | U |  |
| F | U |  |

|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | F | 5 |
| C | F | 3 minimum |
| D | T |  |
| E | U |  |
| F | U |  |

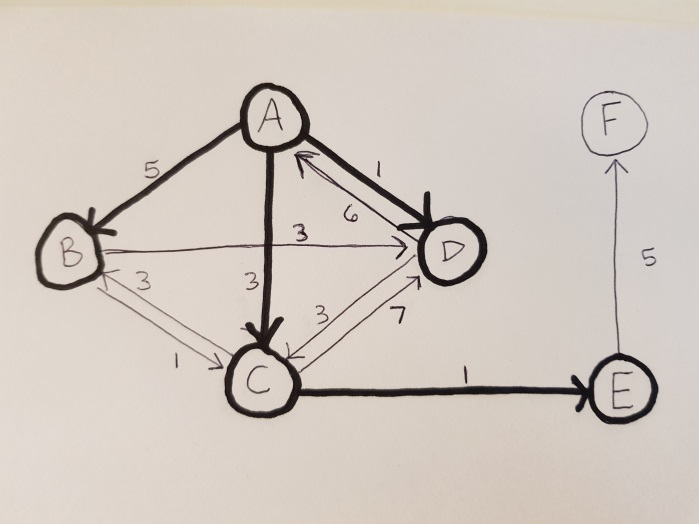




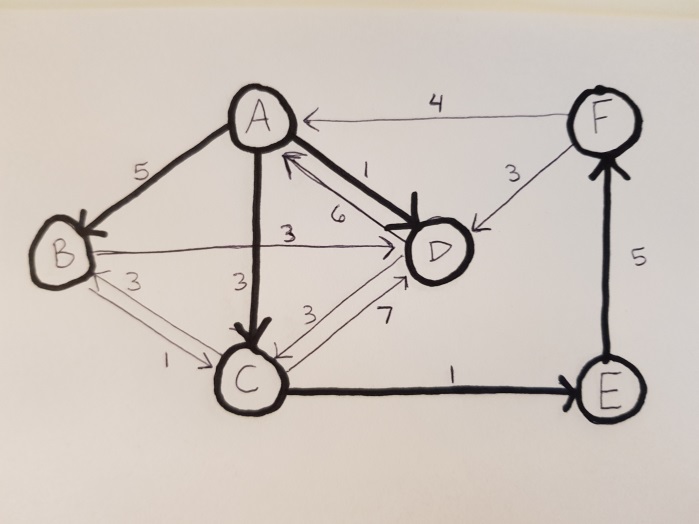
|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | F | 5 |
| C | T |  |
| D | T |  |
| E | F | 4 minimum |
| F | U |  |



|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | F | 5 minimum |
| C | T |  |
| D | T |  |
| E | T |  |
| F | F | 9 |



|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | T |  |
| C | T |  |
| D | T |  |
| E | T |  |
| F | F | 9 minimum |

****

|  |  |  |
| --- | --- | --- |
| A | T |  |
| B | T |  |
| C | T |  |
| D | T |  |
| E | T |  |
| F | T |  |

**Grading Rubric**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Problem** |  | **Meets** |  | **Does Not Meet** |  |
|  |  |  | **10 points** |  | **0 points** |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  | Indicated when the distance from a |  | Did not indicate when the distance |  |
|  |  |  | fringe vertex to the tree was updated |  | from a fringe vertex to the tree was |  |
|  |  | (3) | |  | updated (0) |  |
|  | **Problem 1** |  |  |  |  |  |
|  |  | Indicated which edges became part of |  | Did not indicate which edges became |  |
|  |  |  |  |  |
|  |  |  | the minimum spanning tree and in |  | part of the minimum spanning tree and |  |
|  |  |  | which order (3) |  | in which order (0) |  |
|  |  |  |  |  |  |  |
|  |  |  | Provided the correct final minimum |  | Did not provide the correct final |  |
|  |  |  | spanning tree (4) |  | minimum spanning tree (0) |  |
|  |  |  |  |  |  |  |
|  |  |  | **10 points** |  | **0 points** |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  | Showed what happened each time an |  | Did not show what happened each |  |
|  |  |  | edge was removed from the priority |  | time an edge was removed from the |  |
|  |  |  | queue (3) |  | priority queue (0) |  |
|  | **Problem 2** |  |  |  |  |  |
|  |  | Showed how the dynamic equivalence |  | Did not show how the dynamic |  |
|  |  |  |  |  |
|  |  |  | relation changed on each step (3) |  | equivalence relation changed on each |  |
|  |  |  |  |  | step (0) |  |
|  |  |  |  |  |  |  |
|  |  |  | Provided the correct final minimum |  | Did not provide the correct final |  |
|  |  |  | spanning tree (4) |  | minimum spanning tree (0) |  |
|  |  |  |  |  |  |  |
|  |  |  | **10 points** |  | **0 points** |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  | Provided a correct example of a |  | Did not provide a correct example of a |  |
|  |  |  | weighted graph for which the |  | weighted graph for which the |  |
|  |  |  | minimum spanning tree is unique (2) |  | minimum spanning tree is unique (0) |  |
|  |  |  |  |  |  |  |
|  |  |  | Provided the correct unique minimum |  | Did not provide the correct unique |  |
|  |  |  | spanning tree for that graph (2) |  | minimum spanning tree for that graph |  |
|  |  |  |  | (0) | |  |
|  | **Problem 3** |  |  |  |  |  |
|  |  | Provided a correct example of a |  | Did not provide a correct example of a |  |
|  |  |  |  |  |
|  |  |  | weighted graph that has more than |  | weighted graph that has more than |  |
|  |  |  | one minimum spanning tree (2) |  | one minimum spanning tree (0) |  |
|  |  |  |  |  |  |  |
|  |  |  | Provided two correct distinct minimum |  | Did not provide two correct distinct |  |
|  |  |  | spanning trees for that graph (2) |  | minimum spanning trees for that graph |  |
|  |  |  |  | (0) | |  |
|  |  |  |  |  |  |  |
|  |  |  | Correctly explained what determines |  | Did not correctly explain what |  |
|  |  |  | whether a graph has more than one |  | determines whether a graph has more |  |
|  |  |  | minimum spanning tree (2) |  | than one minimum spanning tree (0) |  |
|  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
|  | **10 points** | **0 points** |
|  |  |  |
|  |  |  |
|  | Clearly indicated which edges became | Did not clearly indicate which edges |
| **Problem 4** | part of the shortest path and in which | became part of the shortest path and |
|  | order (5) | in which order (0) |
|  |  |  |
|  | Provided correct final shortest path | Did not provide correct final shortest |
|  | tree (5) | path tree (0) |
|  |  |  |