Tecnológico de Monterrey

Fundamentos de computación

Homework 8

Student:
Jacob Rivera

Professor:
Dr. Hugo Terashima

May 16, 2019

1 Problems

Solve the following problems:

1. Investigate the algorithm for computing the maximum flow on a graph. Provide and example and apply the algorithm, showing each step.

2. Given a graph $G = (V, E, W)$ and a MST T, suppose that we decrease the weight of one of the edges not in T. Design an algorithm and its computational complexity to find the MST in the modified graph.

   The simplest way would be to add the edge to the MST, which will provoke for a cycle to be created. After that you just have to search for the biggest edge in that cycle and then remove it. As such, the complexity would be $O(n)$, because in the worst scenario, the cycle would include all the edges of the original MST

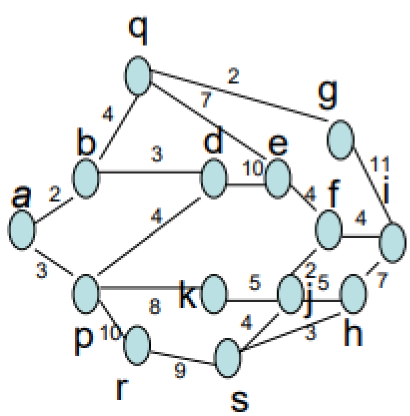

Figure 1: Graph

3. Obtain the MST for the graph in Figure 1 using both the Kruskal and Dijkstra/Prim algorithms.

   Kruskal

   Order the edges

edge	weight
ab	2
ap	3
bd	3
bq	4
de	10
dp	4
ef	4
eq	7
fi	4
fj	2
gq	2
gi	11
hj	5
hi	7
hs	3
jk	5
js	4
kp	8
pr	10
rs	9

edge	weight
ab	2
fj	2
gq	2
ap	3
bd	3
hs	3
bq	4
dp	4
ef	4
fi	4
js	4
hj	5
jk	5
eq	7
hi	7
kp	8
rs	9
de	10
pr	10
gi	11

Construct the MST by adding the unconnected graphs, from lowest to highest by cost.

edge	weight	
ab	2	added
fj	2	added
gq	2	added
ap	3	added
bd	3	added
hs	3	added
bq	4	added
dp	4	not added
ef	4	added
fi	4	added
js	4	added
hj	5	not added
jk	5	added
eq	7	added
hi	7	not added
kp	8	not added
rs	9	added
de	10	not added
pr	10	not added
gi	11	not added

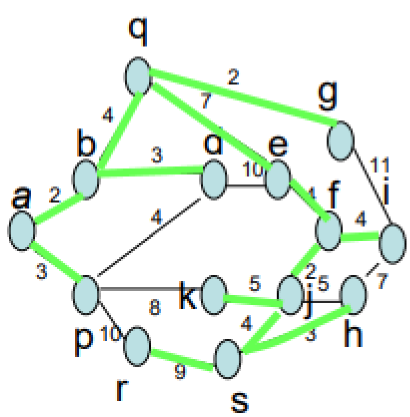

Figure 2: MST by Kruskal

Prim–Dijkstra
Starting conditions:
Start with a

- Tree = $\{a\}$
- Frontier = $\{b, p\}$
- Unvisited = $\{d, e, f, h, i, j, k, q, r, s\}$
- MST = $\{\}$

Final conditions:

- Tree = $\{a, b, d, e, f, h, i, j, k, p, q, r, s\}$
- Frontier = $\{\}$
- Unvisited = $\{\}$
- MST = $\{ab, ap, bd, bq, qg, qe, ef, fj, fi, js, sh, jk, sr\}$

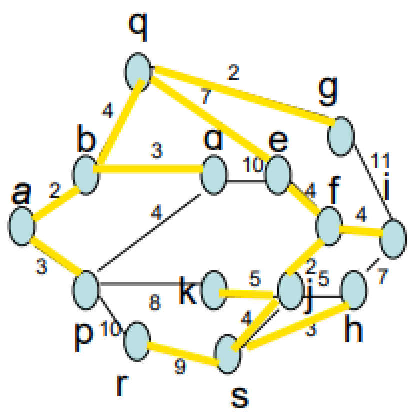

Figure 3: MST by Prim–Dijkstra

4. Design a graph with at least 4 components (biconnected, each with three or more nodes) and run the algorithm seen in class to obtain them. Show the steps.

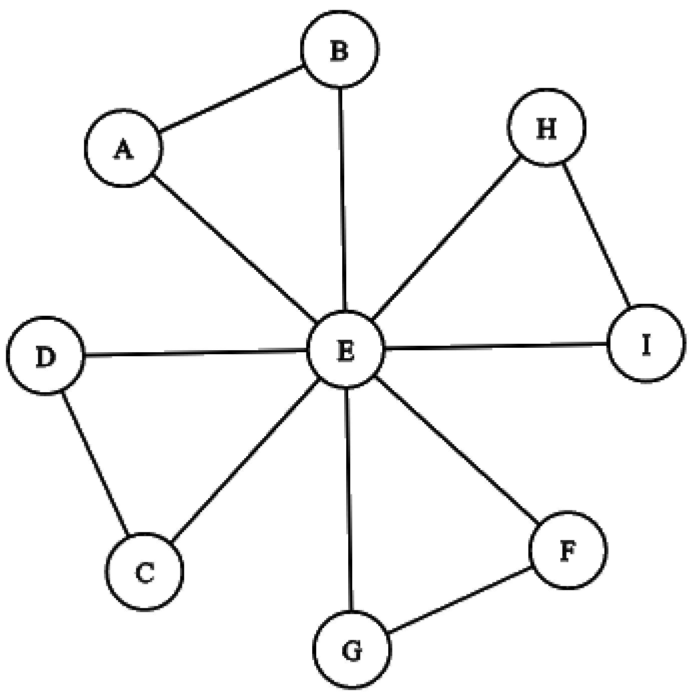

Figure 4: 4 biconnected components graph

- Star with 1
- A → (1, 1)
- B → (2, 2)
- E → (3, 3)
- Find back edge to A
- E → (3, 1), B → (2, 1)

- C $\rightarrow$ (4, 4)
- D $\rightarrow$ (5, 5)
- Find back edge to E
- C $\rightarrow$ (4, 3), D $\rightarrow$ (5, 3)
- F $\rightarrow$ (6, 6)
- G $\rightarrow$ (7, 7)
- Find back edge to E
- F $\rightarrow$ (6, 3), G $\rightarrow$ (7, 3)
- H $\rightarrow$ (8, 8)
- I $\rightarrow$ (9, 9)
- Find back edge to E
- H $\rightarrow$ (8, 3), I $\rightarrow$ (9, 3)

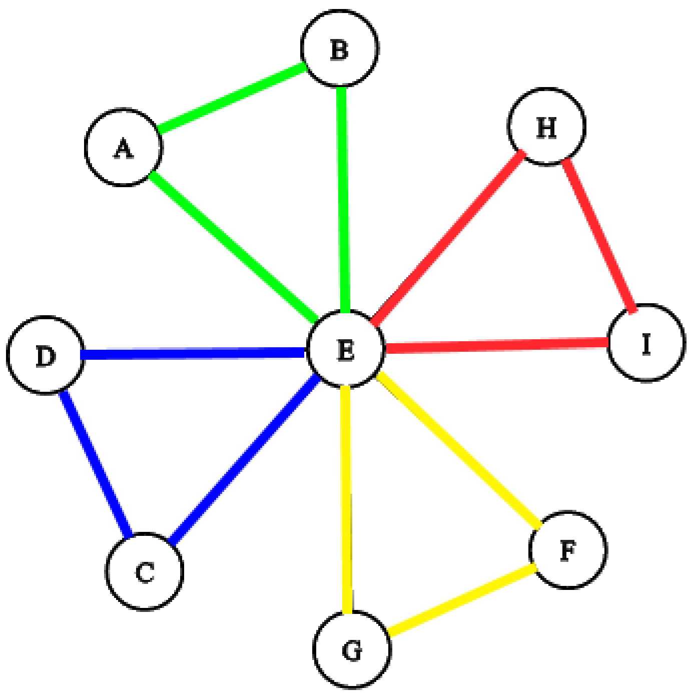

Figure 5: 5 biconnected components graph colored

5. Design a directed graph and establish an origin and a destination and apply the algorithm Dijkstra/Prim to obtain the shortest path between both nodes. Show the steps.

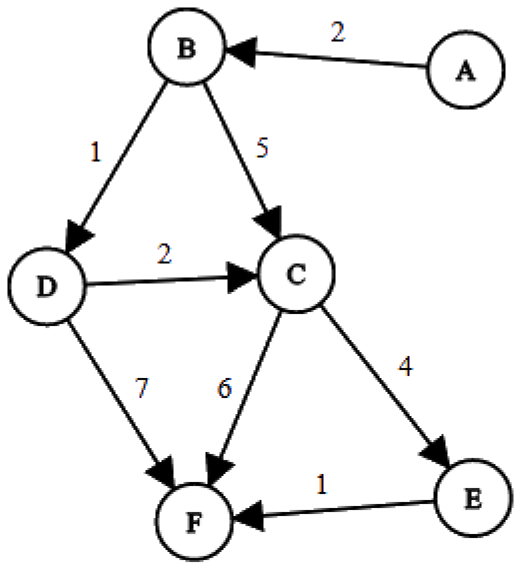

Figure 6: Graph for shortest path

Get a shortest path from A to F

- Open = $\{(A,0)\}$, Frontier=$\{(B,2)\}$, Closed=$\{(C,\infty),(D,\infty),(E,\infty),(F,\infty)\}$
- Open = $\{(A,0),(B,2)\}$, Frontier=$\{(C,7),(D,3)\}$, Closed=$\{(E,\infty),(F,\infty)\}$
- Open = $\{(A,0),(B,2),(D,3)\}$, Frontier=$\{(C,5),(F,10)\}$, Closed=$\{(E,\infty)\}$
- Open = $\{(A,0),(B,2),(D,3),(C,5)\}$, Frontier=$\{(E,9),(F,10)\}$, Closed=$\{\}$
- Open = $\{(A,0),(B,2),(D,3),(C,5),(E,9)\}$, Frontier=$\{(F,10)\}$, Closed=$\{\}$
- Open = $\{(A,0),(B,2),(D,3),(C,5),(E,9),(F,10)\}$, Frontier=$\{\}$, Closed=$\{\}$

Shortest path is $A \to B \to D \to F$, alternate same length route is $A \to B \to D \to C \to E \to F$