CSC 226
Zach Serviss
V00950002

Assignment 4

1.

Init	1	2	3	4	5
1	0	3	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	-5	0	∞
5	∞	∞	∞	6	0

K=1	1	2	3	4	5
1	0	3	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

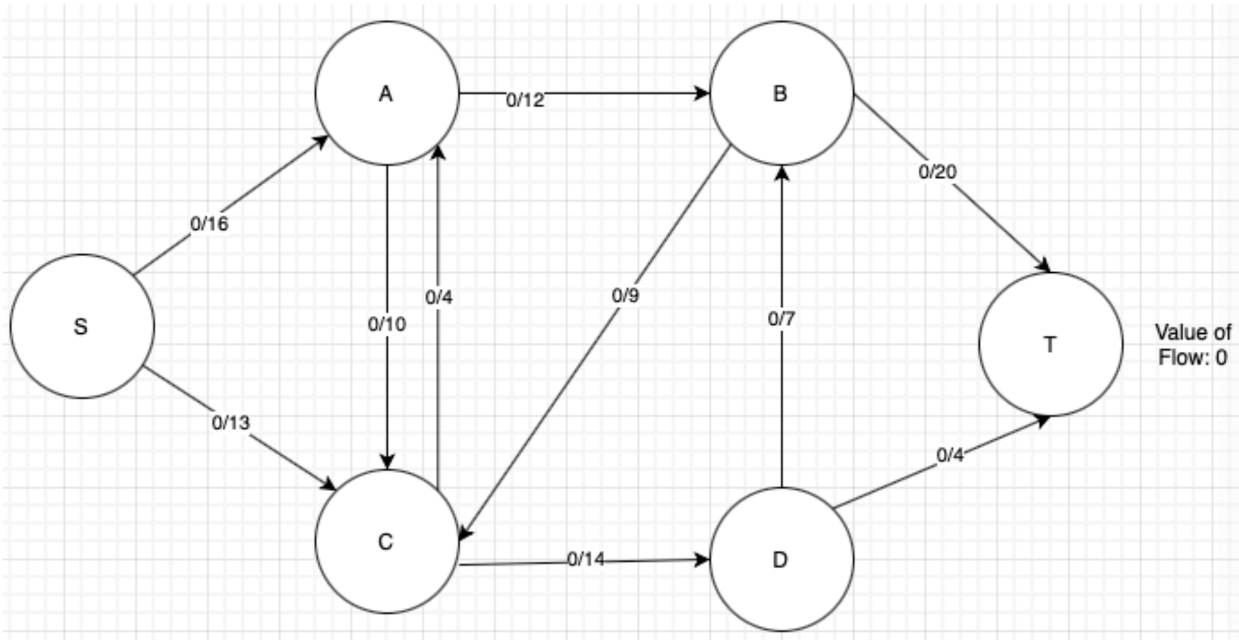
K=2	1	2	3	4	5
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	5	-5	0	-2
5	∞	∞	∞	6	0

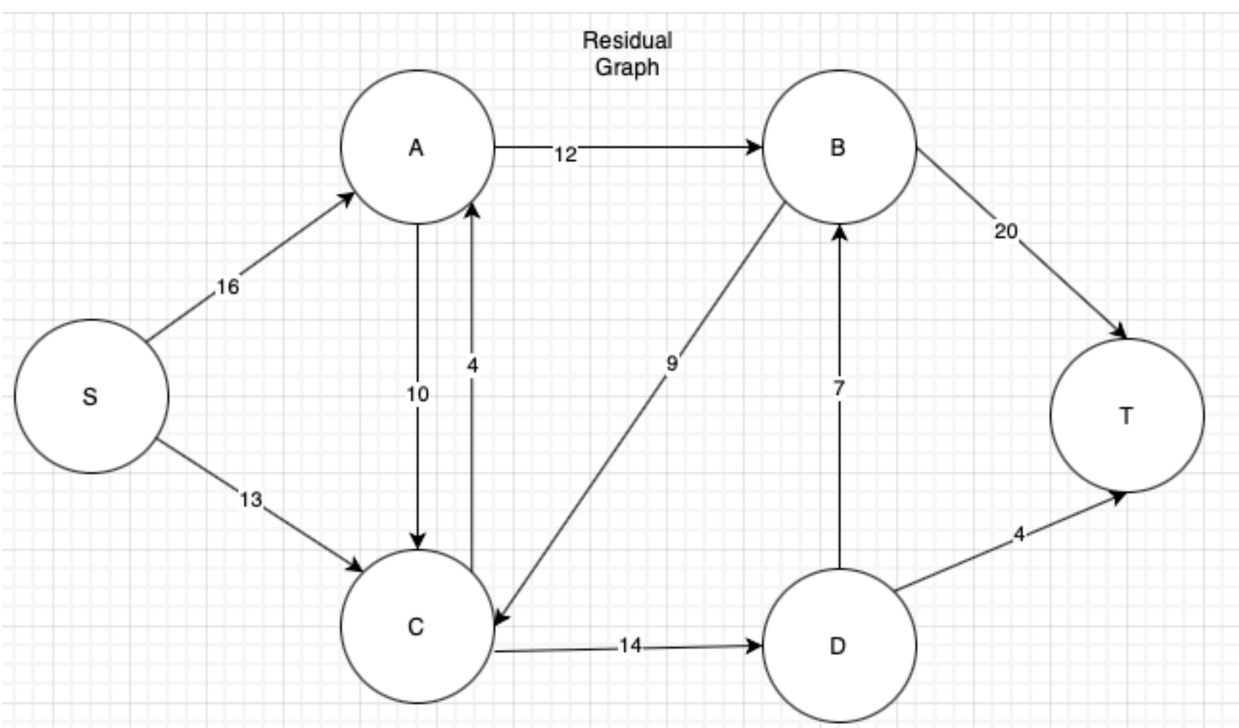
K=3	1	2	3	4	5
1	0	3	8	4	-4
2	∞	0	∞	1	7
3	∞	4	0	5	11
4	2	-1	-5	0	-2
5	∞	∞	∞	6	0

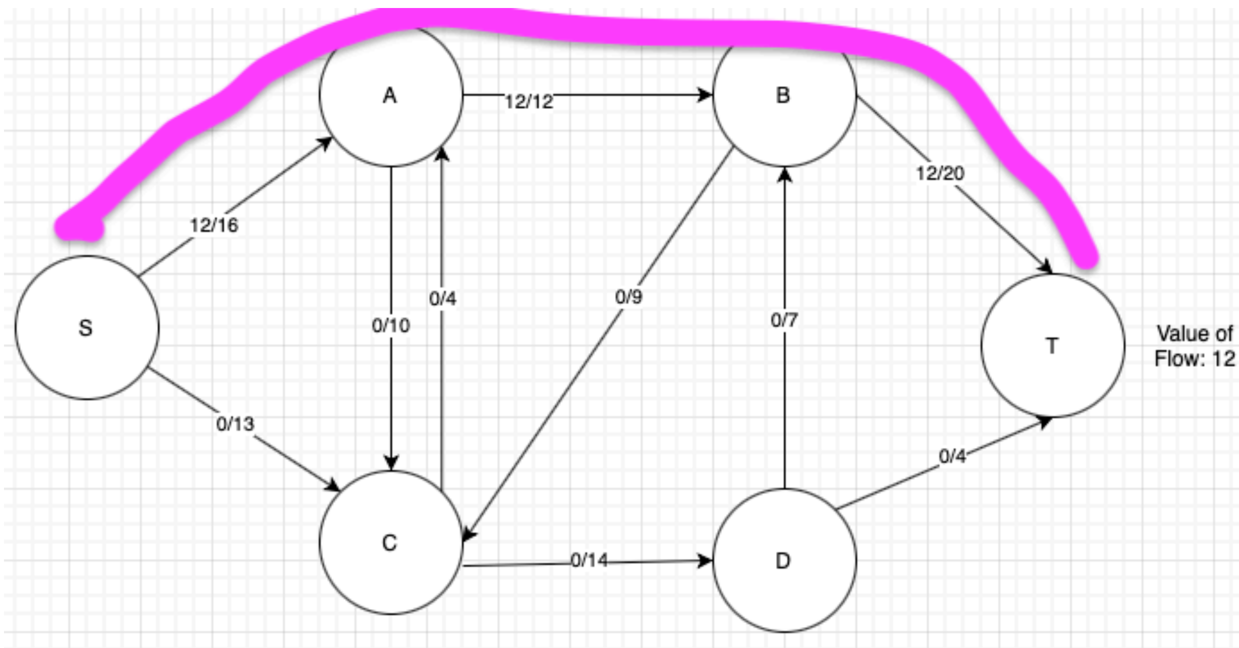
K=4	1	2	3	4	5
1	0	3	-1	4	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

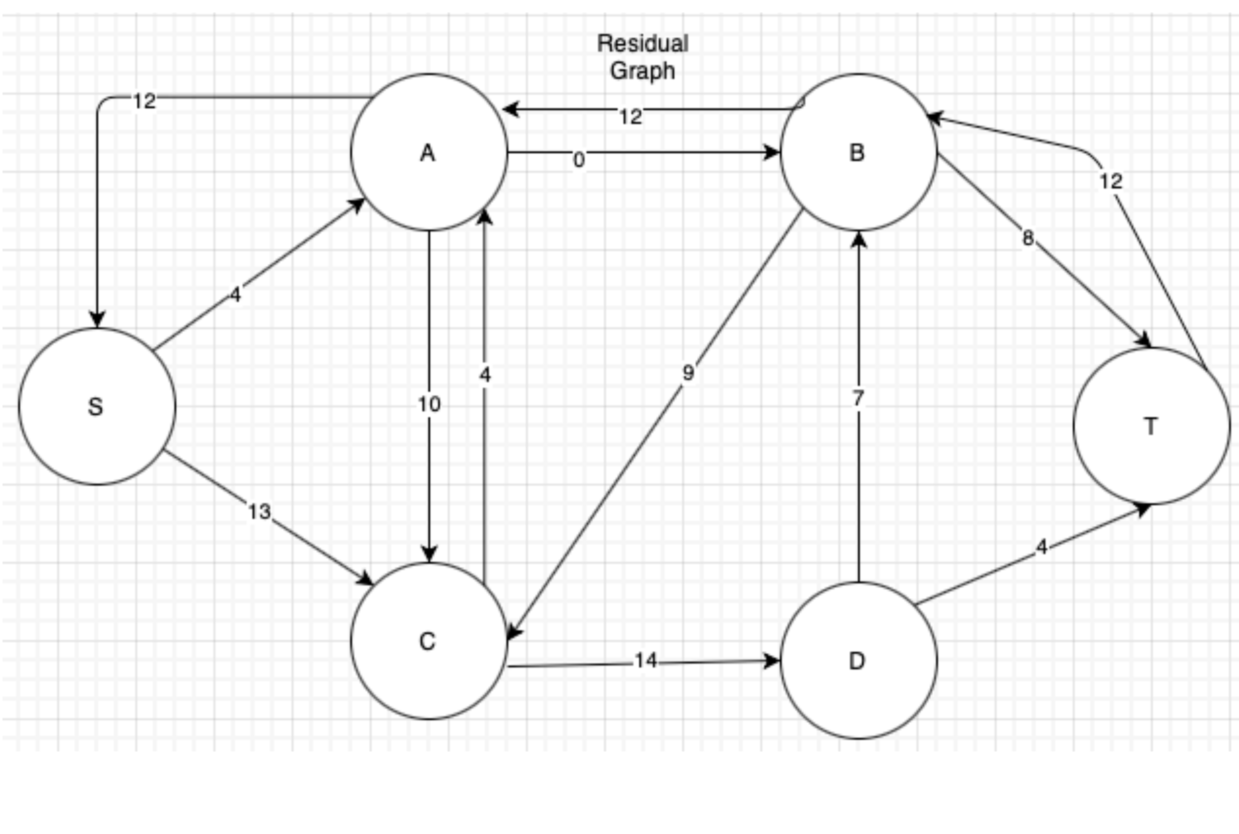
K=5	1	2	3	4	5
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

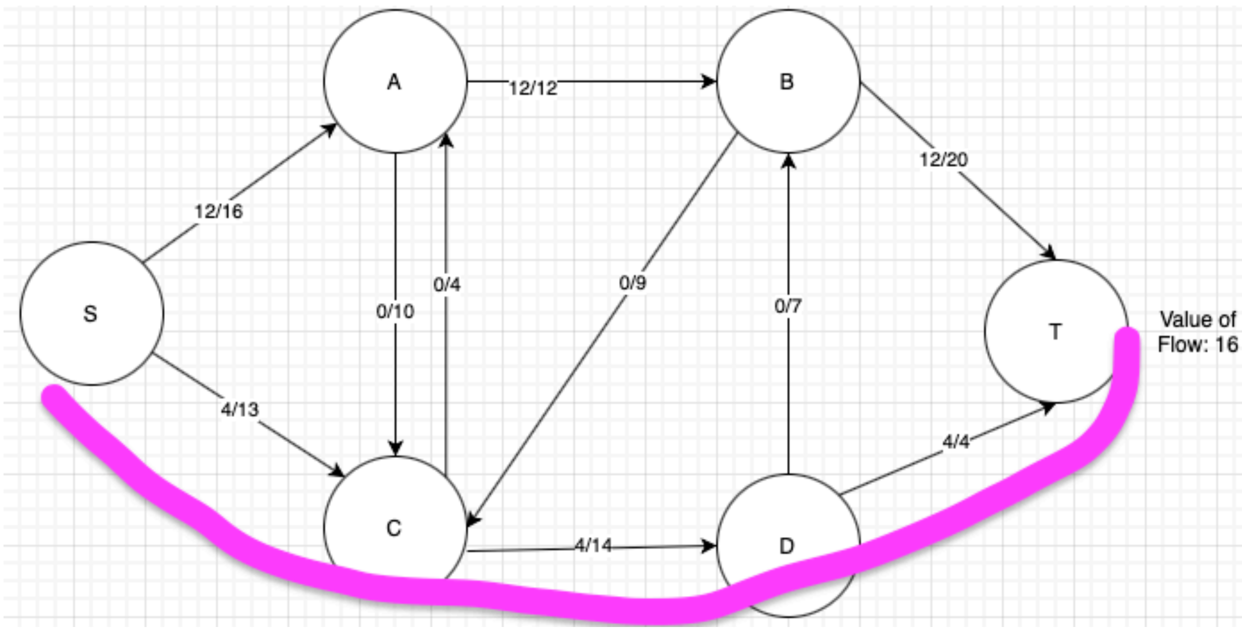
2. The following photos show the Edmunds-Karp algorithm to find the maximum flow of the following graph.

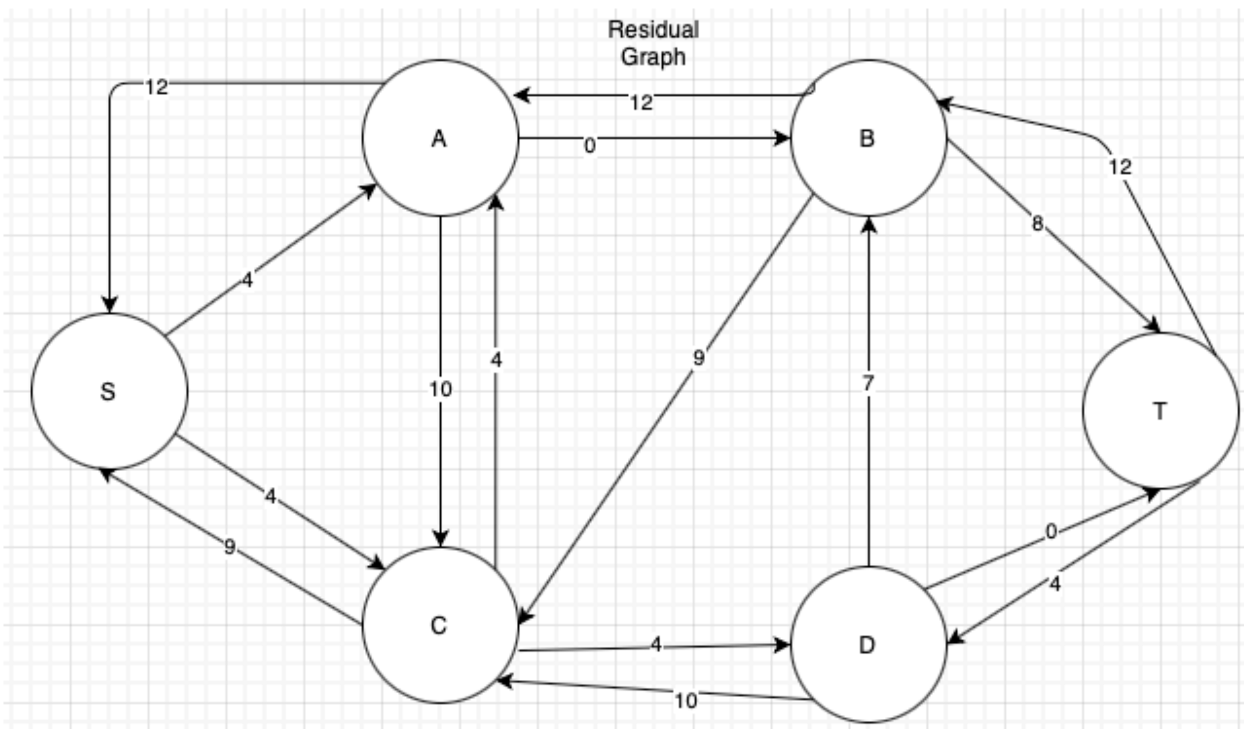


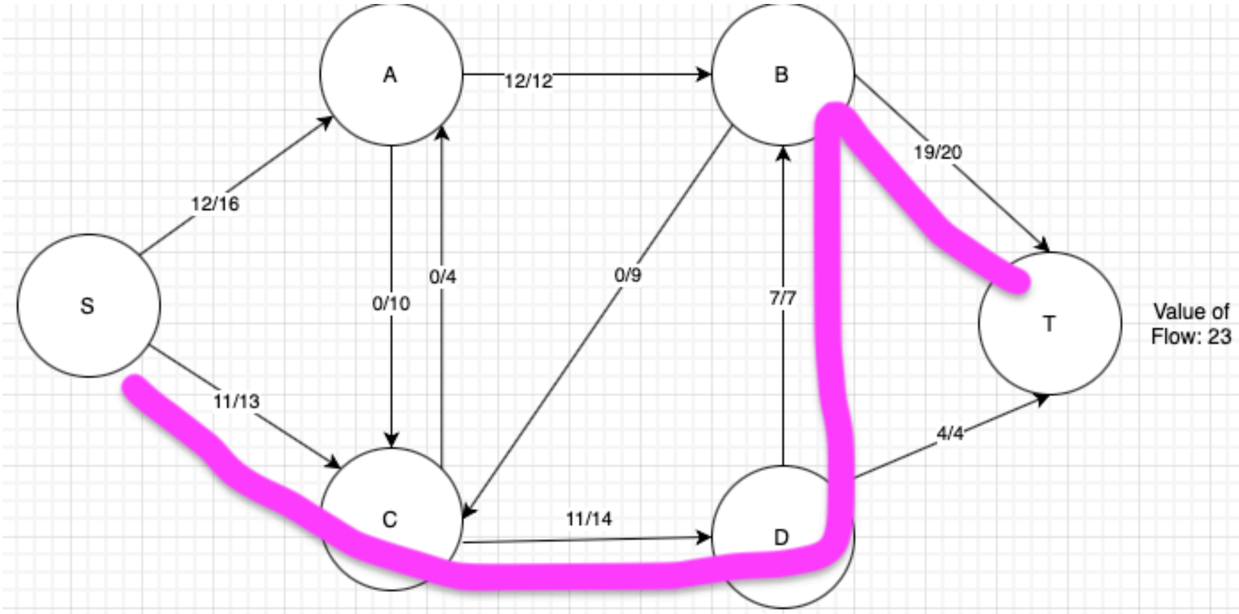


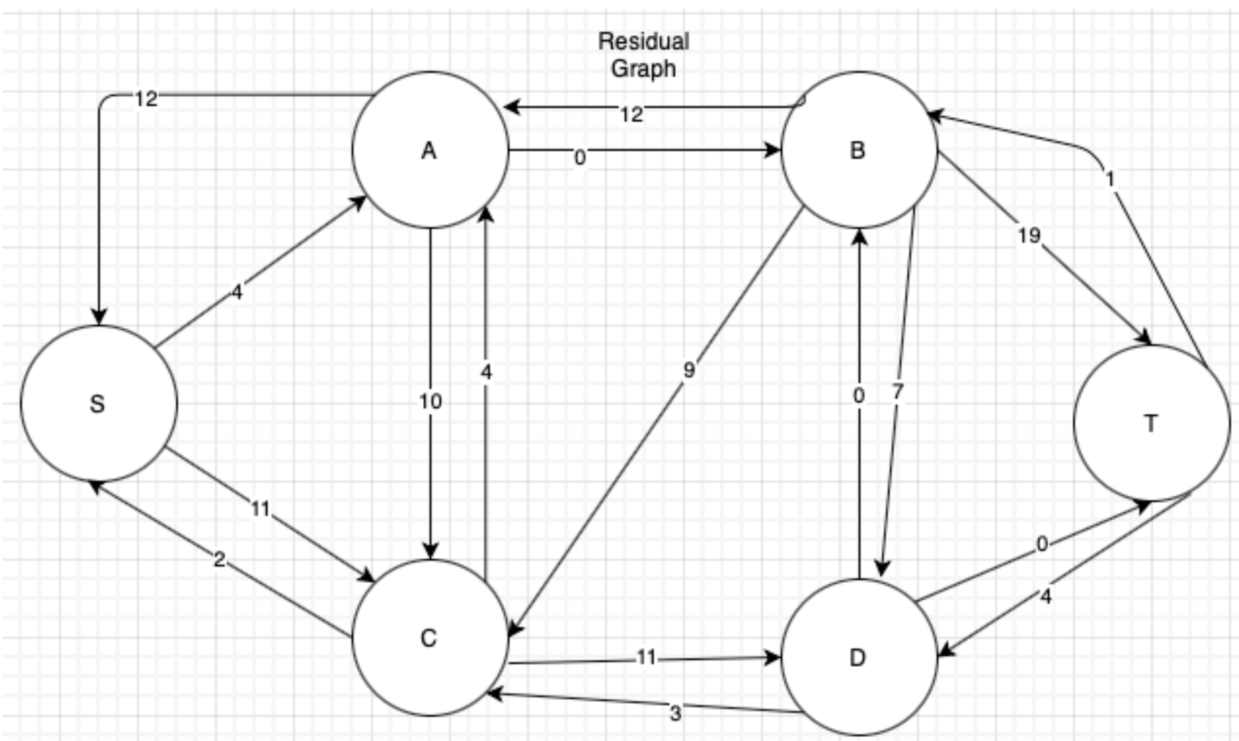

A
B
S
T
C
D
12/12
12/16
12/20
0/10
0/4
0/9
0/7
0/13
0/4
0/14
Value of
Flow: 12


Residual
Graph
A
B
S
T
C
D
12
12
0
12
4
8
10
4
9
7
13
4
14


A
B
S
T
C
D
12/12
12/16
12/20
0/10
0/4
0/9
0/7
4/13
4/4
4/14
Value of Flow: 16


Residual Graph
A
B
S
T
C
D
12
12
0
12
4
8
10
4
9
7
4
9
0
4
4
10





After this point there are no more augmenting paths to be found as we have found the max flow after 3 iterations.

3. If ran in a Ford-Fulkerson method on the adjacency list we will find wether or not the graph given is max-flow or not. All we have to do is find if there are any augmenting paths, if there is an augmenting path, the flow is not max.

For this question we will take the adjacency list and the compare the values of the flow and capacity for each edge. If the value of (flow-capacity) is greater than zero we will add it to a new graph G'. This graph is essentially the residual graph of G.

We will then run a BFS on the new graph G', starting from s and going to t.

If t cannot be reached, the flow is a max flow, if t can be reached it is not a max flow.

The run time for this algorithm will be O(m) because we are only running the BFS once.

4. We will use a modified version of edmonds and karp to find edge-connectivity.

We'll create a digraph G' out of the graph G by adding 2 opposite directed edges and we'll set the capacity of each edge to 1.

Well also set a variable 'min' to the number of edges in G.

We will then choose any vertex s from the graph G', then for all vertices t (s != t), We will then run the edmonds and karp algorithm on each pair of vertices (s and t) in the digraph where s is the source and t is the sink.

After the max flow is calculated, we compare that value to the min variable and assign min the minimum of the two values.

The minimum of these max flows give us the edge-connectivity of the original graph.

The runtime of this algorithm then will be O(V-1) iterations of the edmonds and karp algorithm. And edmonds and karp algorithm runs in O(V E^2). Therefore the total runtime is O(V-1)xO(V E^2) which is O(V^2 E^2 - V E^2). This then reduces down to O(V^2 E^2) (V= number of vertices and E = number of edges).