CS 4820, Spring 2019	**Homework 6, Problem 1**
Name: Kevin Klaben NetID: kek228 Collaborators: Quinn Lui	

(1) *(10 points)* You are trying to sabotage a flow network $G = (V, E)$ on n nodes and m edges by removing at most k edges from G to form a new flow network G'. Assume that every edge in G has unit capacity. Design an algorithm that given as input a flow network G and an integer $k > 0$ outputs a list of at most k edges to be removed from G such that value of the maximum flow of the resulting flow network G' is minimized. Your algorithm should run in time $O(mn)$.

Algorithm:
The first part of this algorithm is to run the Ford-Fulkerson algorithm. Then BFS should be run on the resulting G_f. As discussed in class, all vertices which can be reached from the source side will be the vertex set A of all the vertices on the source side of the min-cut. For each vertex in A simply iterate over all edges and for each edge the goes from the A to the set of vertices V\A should be removed from the graph until k edges have been removed.

Runtime Analysis:
As discussed in class the runtime of the the Ford-Fulkerson is O(mC) where C m is the max flow. As the max flow in this graph due to unit edge capacities can be at most (n-1) due to the fact that at most n-1 edges can come out of the source node corresponding to each other vertex and thus the runtime of Ford-Fulkerson here is O(mn). Following Ford-Fulkerson is a depth first search to find the vertex set of A which will take time O(m+n). Finally an iteration over potentially all m edges will take a max of O(m) time as the edges are removed. Thus, the overall runtime is O(mn+m+n+m)=O(mn).
Proof of Correctness:
Claim:
The algorithm produces the graph with minimized max flow.
Proof:
As shown in class, the Ford-Fulkerson algorithm will produce the max flow within the graph and a residual graph which will have some cut in which no flows can be furthur increased. When a Breath first search is done, all edges reached along this residual graph will be source side nodes of the min cut. All other nodes are sink side nodes of the min-cut. As was proven in class by the min-cut theorem, the max flow is equivalent to the min cut. Thus, as we iterate and remove edges that go from the source side to the sink side, we are removing edges from the min-cut and thus, for each edge removed as the edges have unit capacities, the min-cut will decrease by 1 and thus the max flow remaining will decrease by 1. If k¿ the number of edges i the min-cut then the resulting graph will have a max-flow=0 as all edges in the min-cut will have been removed. Otheriwse the resulting graph will have maxflow = previous maxflow -k. This will minimize max-flow as everyother cut of the original graph must have had a larger capacity than the min-cut as if they had a lower capacity than the min-cut then this cut would have been found by the BFS and thus it would have been the recognized min-cut. As the min-cut capacities are reduced as much as possible in the resulting graph, so is the max flow and thus the claim holds.