5. A new instructor is claiming that to find the max-flow if we use the shortest augmenting paths, then we do not need to consider the residual network. Prove this idea or present a counterexample.

Solution:
I also think that we don't need to use a Residual Graph in the Shortest Path method. The only use of the Residual Graph in the pseudo code given in class is for us to run BFS on to find the shortest path. However, I claim that having back-edges is useless for us and doesn't effect the shortest path of the graph and we can use other graphs to run BFS on.
Claim: back-edges have no use in finding the shortest path of a graph.
Proof: lets assume your BFS algorithm takes an edge and then decides to take the back-edge (the opposite direction of the edge it just took). This can only happen if the BFS took the edge and then reached a dead end so it had to pop back and find another way. So the the edge and the reverse-edge just taken will be essentially a loop that will only increase the length of the path and thus no back-edge will be used in a shortest path.
So by this logic we don't need to use a Residual Graph to run the BFS on since the only thing that a Residual Graph has which is back-edges will not be used. That being said we need to find a new way of keeping track of our flow on the edges since if we run BFS on a normal graph we will always get the same path. I propose that instead of using a Residual Graph we create a copy of the original graph were we put n edges with capacity 1 between every vertex, where n is the capacity of the edge in the original graph. Then we can run BFS on the new graph and when we decide on a flow through a path we just remove that many edges from that path, now we can just run the BFS on one graph and update it as we go along and we don't need a Residual Graph.