Assignment 7 Problem 2

a) Note that for a graph $m \leq n(n-1) \in O(n^2)$

For the instance of problem MAXWEIGHTSIMPLEPATH, for each 3-tuple (u_i, v_i, w_i), the size in bits should be $O(\log u_i + \log v_i + \log w_i)$. Note that $u_i, v_i \in O(n)$ and size of bit of w_i is in $O(\log W)$.

Therefore, the size in bits required to represent an instance of problem MAXWEIGHTSIMPLEPATH should be $O\big(m(2\log n + \log W)\big) = O(n^2 \log(nW))$ since there are m such 3-tuples.

b) Suppose we have a polynomial time algorithm $A_{MAXWEIGHTSIMPPATHDEC}(G, k)$:

On input G with n vertices and threshold weight k, it runs in poly(n) time, and returns True if G has a simple path whose weight is at least k, False is G does not have a such simple path.

Then, we can find value of k_{OPT} by testing all k.

Check edges one by one:

- If $A_{MAXWEIGHTSIMPPATHDEC}(G \backslash e, k) == true$

  This means that there is come optimum that does not use e, so we can remove it
- Otherwise, all optimum subsets use e

  So, keep the edge in the graph.

After doing this, what is left is a heaviest simple path, since we check all edges and removes the edges that is not needed.

Note that we are calling polynomial times of $A_{MAXWEIGHTSIMPPATHDEC}$, since k can be at most $mW \in O(n^2W)$. So finding k_{opt} calls $O(n^2W)$ times. To test all edges, there are $O(n^2)$ edges in total, and for each edge, we call decision algorithm once, therefore, testing edges requires $O(n^2)$ calls of decision algorithm.

Also, note that we can construct the instance of decision problem in polynomial time. From part a, we know that the instance of optimization problem is $O(n^2 \log(nW))$, which is polynomial. To modify G into G\e, polynomial time is sufficient, by searching edge e and remove it from the instance. Assume that G' is the graph at the end of the algorithm. Suppose $e' \in G$ and e' is not in the heaviest path. Then when we doing algorithm, we checking on e', The heaviest path we will finally get and e' are both exist in the graph, then if we applying decision algorithm on G\e', the result should be true and we should remove e' from the graph. Which contradicts our assumption. Therefore, all edges rested are in the heaviest path that we desire.

So, if there is a polynomial time algorithm for MAXWEIGHTSIMPPATHDEC, then there is one for MAXWEIGHTSIMPPATH.

Therefore, $MAXWEIGHTSIMPPATH \leq_p MAXWEIGHTSIMPPATHDEC$