CS302 - Final Review

1. Given two strings x and y, create a dynamic programming algorithm to compute their *optimal edit distance*, where the edit distance the numbers of insertions, deletions, substitutions, or transpositions (switching the order of two adjacent letters), in some sequence that changes x into y. The optimal edit distance is the smallest possible edit distance. For example, the optimal edit distance of SHOAL and COLA is 3, as we can remove S, changing $H \to C$, and transposing the L and the A. Edit distance is useful for spell checking applications and genomic applications.

2. DOUBLE-k-INDSET is the problem whose input is a graph, and the output is YES if and only if there are at least 2 distinct independent sets of size at least k in the graph. By distinct, I mean that the intersection of the two independent sets is empty. Prove DOUBLE-k-INDSET is NP-Complete.

3. What is the average runtime of randomized search without replacement if there are c copies of the item we are looking for out of an array of size n. Please use indicator random variables and go through the usual routine. You do not need to simplify your final answer - it can be a messy sum.

4. For the following statements regarding Dijkstra's algorithm, either explain why it is true (formal proof not required), or provide a counter example.

   (a) Consider a graph G that is directed, has negative edge weights, but no negative cycles (a negative cycle is a cycle where the sum of edge-weights in the cycle have negative value.) Then there will always be a vertex where the incorrect distance is calculated.

   (b) Consider a graph G that is directed, and that has a negative cycle that is reachable from s. Then there will always be a vertex where the incorrect distance is calculated.

5. Return to conference scheduling: Suppose you have n events, each with a start time s_i and end time f_i, for $i \in \{1, \ldots, n\}$. Unfortunately, you only have one auditorium, and you can't schedule conflicting events (events where a start time of one is between the start time and end time of another.) You would like to maximize the number of events that are held. Consider an algorithm that at each iteration, picks the remaining event with the earliest finish time. Prove this algorithm is optimal.