Emily Crilley

Homework Six

1. This approach is a greedy algorithm, even if it goes from the end to the beginning. This is because a greedy algorithm looks for the best option in the current moment, searching for the optimal solution. This is the same as the original problem would be, just running in reverse. The same logic would apply, where we are searching for the most optimal choice at each step.
2.
   a. Least Duration

Activity	Start	Finish	Duration
1	3	5	2
2	1	4	3
3	4	7	3

The solution for this approach is {1} (activity 1), but the optimal solution should be {2,3} (activity 2,3).

   b. Fewest Overlaps

Activity	Start	Finish	Number of overlaps
1	0	1	2
2	1	2	2
3	1	3	3
4	1	3	3
5	2	4	3

The solution for this approach is one of {1,2} but the optimal solution is {2,3,4,5}.

   c. Earliest Start Times

Activity	Start	Finish
1	1	5
2	2	3

3	4	5

The solution for this approach is {1} but the optimal solution should be {2,3}

3. 1: If certificate y is permutation of {1,2,...,n} continue, else return false

   2: Permute vertices of G1, verify the permuted G1 is identical to G2

   Step 1 takes O(V^2) time, Step 2 takes O(V + E)

   Algorithm verification runs O(V^2) meaning GRAPHISOMORPHISM $\in$ NP

4. Assume P = NP

   For every X $\in$ NP, X $\in$ P and P are closed under complement, so $\overline{X} \in$ P and therefore X $\in$ coNP

   For every X $\in$ coNP, $\overline{X} \in$ P and P are closed under complement, so X $\in$ P and therefore X $\in$ NP

   If P = NP, the polynomial time hierarchy collapses to the lowest level and this implies that P = NP = coNP = PH

   Due to the contrapositive proof, P = NP $\Rightarrow$ NP = coNP and this implies that NP != coNP then P != NP

5. The truth table proves that ψ is not satisfiable

x1	x2	x3	x1 $\vee$ x2	$\wedge$ x3	$\wedge$	x1 $\wedge$ x2	$\wedge$	$\overline{X}3$	$\vee$ x3	$\wedge$	x1 $\wedge$ x2	$\wedge$	$\overline{X}3$
T	T	T	T	T	T	T	F	F	T	**F**	T	F	F
T	T	F	T	F	F	T	T	T	T	**F**	T	T	T
T	F	T	T	T	T	F	F	F	T	**F**	F	F	F
T	F	F	T	F	F	F	F	T	F	**F**	F	F	T
F	T	T	T	T	T	F	F	F	T	**F**	F	F	F
F	T	F	T	F	F	F	F	T	F	**F**	F	F	T

6. A propositional formula has a truth value. Looking at the value as a boolean, as the values are true/false, the satisfiability can be seen to be true if the clause is evaluated to 1, as any of its clauses that produces a 1 will automatically make the clause true. This would be run in polynomial time, as the number of clauses would determine the time.

7.

a. The time complexity of the algorithm would be O(nW), where n = number of items in the knapsack and W = max weight it can hold.
b. P = NP, as the knapsack problem is NP-hard, it would mean that unless P = NP, a polynomial time would be impossible.