CSCI 665 Assignment 6

1.
Since we are performing the operations in a reverse order , this will be similar to the original activity selection problem when time runs in reverse. Hence, it produces the optimal results for the same set of reasons. It is greedy because we make the best looking choice at each step.

2.
As a counterexample to the optimality of greedily selecting the shortest,
suppose our activity times are {(1, 5),(8, 15),(10, 18)} then, picking the shortest
first, we have to eliminate the other two, where if we picked the other two
instead, we would have two tasks not one.
Consider the following example of activity times :
{(−1, 0),(2, 6),(0, 3),(0, 3),(0, 3),(4, 7),(6, 9),(8, 11),(8, 11),(8, 11),(10, 12)}. Then,
by this greedy strategy, we would first pick (4, 7) since it only has a two conflicts. However, doing so would mean that we would not be able to pick the
only optimal solution of (−1, 1),(2, 5),(6, 9),(10, 12).
As a counterexample to the optimality of greedily selecting the earliest start
times, suppose our activity times are {(1, 10),(2, 3),(4, 5)}. If we pick the earliest start time, we will only have a single activity, (1, 10), whereas the optimal
solution would be to pick the two other activities.

3.
Consider 2 graphs G1 and G2 and certificate z be the indices {i1,i2,i3…….in}
The following pseudocode verifies whether a graph is Isomorphic or not:
A(G,z)

1. If z is a permutation of 1,2,3…..n.
   If true:-return false
   Else continue
2. Permute the vertices of G1 of given permutation and check if G1 is identical to G2.

Step 1 takes at most O(V^2) time and step 2 runs in O(V+E) time. Therefore, the verification algorithm A runs in O(V ^ 2) time. Hence, GRAPHISOMORPHISM ∈ NP.

4.
Proving by Contradiction:
Hence,
Consider P=NP
Conversely, we can say that NP=co-NP . Since P is closed under complementation , NP is also closed under complementation for the same case.
Therefore, if NP≠ co-NP, then P ≠ NP.

5.

P	Q	R	(((P V Q) ⋀ R)⋀(((P⋀(Q⋀~R))VR)⋀(P ⋀ (Q ⋀~R))))
F	F	F	F

F	F	T	F
F	T	F	F
F	T	T	F
T	F	F	F
T	F	T	F
T	T	F	F
T	T	T	F

Since the relation is Fallacy , ψ is not satisfiable.

6.
DNF is composed of clauses ORs and ANDs. The DNF is satisfiable only if any of its clauses can be evaluated to 1. If one of the clauses contains a variable x in the form x /\ ~x, the clause will evaluate to 0 whatever the Boolean value x is assigned ; else there is some combinations of 0 & 1 that will evaluate to 1.
Hence, the algorithm will return false if any of the clauses contains a variable x and ~x in the same clause. Else it will return true.If there are m clauses and n literals at most in each clause. Each clause can be evaluated in O(n^2) time. The time complexity of the whole algorithm would be O(mn^2)

7.
a.
O(nW). where n is the number of items and W is the capacity of knapsack.

b.
The size of W is dependent on the number of bits to store which is exponential. The algorithm is a pseudo polynomial time algorithm. Hence, the analysis does not argue that the knapsack problem is NP-complete.

8.
Following are the conditions to label a problem A to be NP-complete.
(1) there is a non-deterministic polynomial-time algorithm that solves A,
(2) any NP-Complete problem B can be reduced to A,
(3) the reduction of B to A works in polynomial time,
(4) the original problem A has a solution if and only if B has a solution.

Condition 1:
Predict 2 partitions and these 2 have the same sum.

Condition 2:
We can reduce Subset-Sum problem to Partition problem as follows.
Let X be a set of integers and t be a target.
We find a subset Y of X which contains members which add up exactly to t.
Let s be a sum of integers of X.

We input Partition :-X` where X U {s-2t}
We accept only if Partition accepts.

Condition 3:
This reduction works in polynomial time.

Condition 4:
(X,t) ∈ Subset-Sum if and only if X` ∈ Partition

Hence Partition is NP complete.