1) 20 pts
Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[TRUE/FALSE] True
Assume P !=NP. Let A and B be decision problems. If A is in NP-Complete and $A \leq_P B$, then B is not in P.

[TRUE/FALSE] True
There exists a decision problem X such that for all Y in NP, Y is polynomial time reducible to X.
It is in fact the Cook-Levin theorem that proves that there are problems that are NP-complete. Picking X to be 3-SAT (or any other in problem in NP-complete) ensures that ensures that every Y in NP is reducible to X.

[TRUE/FALSE] True
If P equals NP, then NP equals NP-complete.
True. A problem X is NP-hard iff any problem in NP can be reduced in polynomial time to X. If P equals NP, then we can reduce any problem in NP to any other problem by just solving the original problem.

[TRUE/FALSE] False
The running time of a dynamic programming algorithm is always theta(P) where P is the number of sub-problems.
Solution: False. The running time of a dynamic program is the number of subproblems times the time per subproblem. This would only be true if the time per subproblem is O(1).

[TRUE/FALSE] True
A spanning tree of a given undirected, connected graph G=(V,E) can be found in O(| E|) time. You can just walk on the graph.

[TRUE/FALSE] True
To find the minimum element in a max heap of n elements, it takes O(n) time

[TRUE/FALSE] True
Kruskal's algorithm for finding the MST works with positive and negative edge weights.

[TRUE/FALSE] False
If a problem is not in P, then it must be in NP.

[TRUE/FALSE] False
If an NP-complete problem can be solved in linear time, then all NP-complete problems can be solved in linear time.

[TRUE/FALSE] True