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

[TRUE/[✓ FALSE]]
Let X be a decision problem. If we prove that X is in the class NP and give a poly-time reduction from X to 3-SAT, we can conclude that X is NP-complete.

[TRUE/[✓ FALSE]]
Let A be an algorithm that operates on a list of n objects, where n is a power of two. A spends $\Theta(n^2)$ time dividing its input list into two equal pieces and selecting one of the two pieces. It then calls itself recursively on that list of n/2 elements. Then A's running time on a list of n elements is O(n).

[[✓ TRUE]/FALSE]
If there is a polynomial time algorithm to solve problem A then A is in NP.

[TRUE/[✓ FALSE]]
A pseudo-polynomial time algorithm is always slower than a polynomial time algorithm.

[TRUE/[✓ FALSE]]
In a dynamic programming formulation, the sub-problems must be non-overlapping.

[[✓ TRUE]/FALSE]
A spanning tree of a given undirected, connected graph G = (V, E) can be found in O(E) time.

[[✓ TRUE]/FALSE]
Ford-Fulkerson can return a zero maximum flow for flow networks with non-zero capacities.

[[✓ TRUE]/FALSE]
If $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$, then $h(n) = \Theta(f(n))$

[[✓ TRUE]/FALSE]
There is a polynomial-time solution for the 0/1 Knapsack problem if all items have the same weight but different values.

[TRUE/[✓ FALSE]]
If there are negative cost edges in a graph but no negative cost cycles, Dijkstra’s algorithm still runs correctly.