It is recommended that you read through the exam before you begin. Answer all questions in the space provided.

Name: ____________________

Answer whether the following statements are true or false and briefly explain your answer in the space provided.

1. [TRUE / FALSE] Any algorithm with a pair of nested loops has a complexity of $\Omega(n^2)$. [5 pts]

2. [TRUE / FALSE] Because the Travelling Salesman problem is NP-Hard, no large instances, e.g. millions of cities, can be solved quickly. [5 pts]

3. [TRUE / FALSE] A Brute-force solution is always less efficient than a divide and conquer solution to the same problem. [5 pts]

4. (a) What is the best case space complexity of Breadth-first Search? (You do not need to prove this, just briefly describe your reasoning.) [5 pts]

   (b) What is the worst case space complexity of Breadth-first Search? (You do not need to prove this, just briefly describe your reasoning.) [5 pts]

5. The Knapsack problem is NP-Hard, which means that a poly-time solution for it would prove that $P = NP$. Explain why the $O(nC)$ dynamic programming we showed in class, where n is the number of items and C is the capacity of our sack, does not prove that $P = NP$. [5 pts]

6. (a) Briefly explain the difference between Memoization and Tabulation with regards to dynamic programming. [5 pts]

   (b) When would memoization result in a faster run time than tabulation? [5 pts]

   (c) When would tabulation result in a faster running time? [5 pts]

7. There is no known Greedy strategy that is optimal for solving the 0/1 Knapsack problem. For each of the following strategies give a counterexample, i.e. descibe an instance where that strategy will fail to produce an optimal result.

   (a) Lightest item first. [5 pts]

   (b) Most valuable item first. [5 pts]

   (c) Item with the best value to weight ratio first. [5 pts]

Choose just one of the following questions. Circle your chosen question and write your solution on the following pages. Be sure to specify the parameters you are using to measure the input size and identify the basic operation, and briefly explain how your algorithm works. Your answer will be graded for correctness, clarity, and efficiency.

8. Given a directed weighted graph, design an algorithm that determines whether that graph contains a cycle. Find the time and space complexity of your algorithm. [40 pts]

9. Write an algorithm that implements the word wrap operation. It should take a string and an integer representing the screen width. The output should be a list of strings, each representing a line of text. Make sure your algorithm doesn't split any words and correctly handles new line characters that are encountered. Find the time and space complexity of your algorithm. [40 pts]

10. Suppose you are investing. You want to buy low and sell high to maximize your profit. Thanks to the magic of time travel you have acquired an array of future prices, but can only perform two trades, one buy and one sell. Your array of prices is in chronological order and your buy order must precede your sell order. Design a $O(n \log n)$ divide and conquer algorithm for finding the maximum profit you can make. Find the space and time complexity of your algorithm. [40 pts]

11. Design an algorithm that takes an array of integers and finds the maximum product of a triplet in the array. Find the time and space complexity of your algorithm. [40 pts]
    Example:
    Input: [10, 3, 5, 6, 20] Output: 1200 Multiplication of 10, 6 and 20

    Input: [-10, -3, -5, -6, -20] Output: -90

    Input: [1, -4, 3, -6, 7, 0] Output: 168