CISC-365*
Test #1
October 3, 2008

Student Number (Required) ______________________

Name (Optional)________________________________

This is a closed book test. You may not refer to any resources.

This is a 50 minute test.

Please write your answers in ink. Pencil answers will be marked, but may not be reconsidered after the test papers have been returned.

The test will be marked out of 50.

Question 1	/10
Question 2	/24
Question 3	/16
TOTAL	/50

Question 1 (10 marks)

Use induction to show that n^2 is in $O(2^n)$.

Question 2 (24 marks)

Recall the Subset Sum problem: Given a set of n integers **S** and a target integer **k**, does S contain a subset that sums to k? We know that this problem is NP-Complete. In this question you will be asked to classify three variations of this problem. Each part of this question is independent of the others. **For each part, either show that the problem is NP-Complete or explain how you can solve it in polynomial time. (Note: if you can show the problem is NP-Complete AND you can solve it polynomial time, congratulations!) Each part is worth 8 marks.**

(a) Approximate Subset Sum: Given a set of n integers S, a target integer k, and an integer c, does S contain a subset with sum in the range [k-c .. k+c]?

(b) Within 10 Subset Sum: Given a set of n integers S and a target integer k, does S contain a subset with sum in the range [k-10 .. k+10]? (Hint: what happens if you multiply all the values in the set S, and k, by 100?)

(c) 100 Value Subset Sum: Given a set of 100 integers S and a target integer k, does S contain a subset that sums to k? (You won't need a full page to answer this question – the pagination just worked out this way.)

Question 3 (16 marks)

Let A[0..n-1] be an array of integers with the following properties:

- all the integers in A are distinct (no repetitions)
- the integers increase in value from A[0] up to some A[m], then decrease in value to A[n-1]
- either the increasing side or the decreasing side (or both) may be empty – see the examples

As examples, A might look like {3 6 19 18 13 2 -5} or {7 5 1} or {2 8 13}

Create an algorithm to find the value of m, the index of the largest value in A. Note that this is not difficult to do in O(n) time: we can simply look at each value in A and remember the location of the largest. Your solution must run in O(log n) time.

(a) Express your algorithm in clear pseudocode (or in some standard programming language, if you prefer)

(b) State and explain a recurrence relation that describes the running time of your algorithm.

(c) Solve the recurrence relation to show that your algorithm runs in O(log n) time.

Special Bonus Question: (0 marks)

TTTT
WINKS

What is the meaning of the figure above?