CISC/CMPE-365*
Test #2
October 21, 2016

Student Number (Required) ______________________

Name (Optional)________________________________

This is a 50 minute test.

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

The test will be marked out of 50.

Question 1	/12
Question 2	/25
Question 3	/12
Question 4	/1
TOTAL	/50

Question 1 (12 Marks)

(a) **[6 marks]** Show the bitstring codes that result from applying the Huffman Coding algorithm to a string containing the following set of letters with the indicated frequencies:

a	b	c	d	e	f	g	h
1	1	2	3	5	8	13	21

(b) **[3 marks]** Is your answer in (a) unique? Why or why not?

(c) **[3 marks]** Generalize your answer from (a) to describe an optimal prefix-property code when the letter frequencies are the first **n** Fibonacci numbers.

Question 2 (25 marks)

Suppose you have **K** dollars in your pocket, and you want to buy Hallowe'en candies to give to trick-or-treaters. At the candy shop there are **n** small buckets of different types of candy – each bucket has < **K** candies in it. Each piece of candy is priced at $1, so you can only buy a maximum of K pieces of candy. **For each type of candy, you have a satisfaction value** that you experience from giving one piece of that candy to a trick-or-treater.

(a) **[10 marks]** Suggest a Greedy Algorithm to **maximize the total satisfaction** you will experience when you give away all the candy that you buy.

(b) **[5 marks]** Prove that your algorithm's first choice is optimal (i.e. that there is an optimal solution that makes the same choice)

(c) **[10 marks]** Complete the proof that your algorithm finds an optimal solution to the problem

Question 3 (12 Marks)

Is Dijkstra's Algorithm for finding least-weight paths in a graph with positive edge-weights a Greedy Algorithm?   Why or why not?

Question 4 (1 mark)

[Note for 2019 students: we have not covered CNF-SAT so this particular question would not be reasonable for this year's test. I have left it here as an example.]

Consider the following Greedy Algorithm for CNF-SAT:

> sort the boolean variables in the expression in descending order based on how many terms they occur in
>
> for each boolean variable, set it to True unless its negation has already been set to True

True or false: If E is a satisfiable expression in CNF form, this algorithm will always find a truth assignment that satisfies E

true ***FALSE***