COMPSCI 320SC 2019 Midterm Test

University ID: ______________________________

Student Name: ______________________________

Student Signature: ______________________________

Time Finished: ______________________________

Attempt *all* questions. (Use of calculators is NOT permitted.)

Put the answers in the space below the questions.

Write clearly and *show all your work*!

Marks for each question are shown just before each answer area.

This 60 minute test is worth 10% of your final grade for the course.

Good luck!

Question #:	1	2	3	Total
Possible marks:	10	10	10	30
Awarded marks:				

1. Basic Analysis **[10 marks]**

   (a) Show that $\frac{\ln n}{\sqrt{n}}$ is $O(\sqrt{n})$. (*3 marks*)

   (b) Prove that if f_1 is $\Omega(g_1)$ and f_2 is $\Omega(g_2)$, then $f_1 + f_2$ is $\Omega(g_1 + g_2)$. (*3 marks*)

   (c) Show that if $f(n) = \left(\frac{n}{2}\right)^n$ then $f(n) = 2^{\Theta(n \log n)}$. (*4 marks*)

2. Divide and Conquer **[10 marks]**

(a) Master theorem: Consider the following "divide-and-conquer" function:

```
function printer(int n)
  for i = 1 to n do
    for j = i + 1 to n do
      print  CS320
    end for
  end for
  if n > 0 then
    for i = 1 to 4 do
      printer(⌊n/2⌋)
    end for
  end if
```

Let $T(n)$ denote the number of CS320 generated by a call of *printer*(n).

i. Provide a recurrence equation for $T(n)$. (***3 marks***)

ii. Solve the recurrence asymptotically for general n. (***2 marks***)

You may want to make use of the following *master recurrence theorem*: Assume $T(n) = aT(n/b)+g(n)$, where $g(n)$ is $O(n^c)$, is the total time for a divide and conquer algorithm. Then:

$$T(n) = \begin{cases} O(n^c), & \text{if } a < b^c \\ O(n^c \log n), & \text{if } a = b^c \\ O(n^{\log_b a}), & \text{if } a > b^c \end{cases}$$

(b) Finding the peak item in array

A peak item in an array is the item that is greater than its neighbors. If there are more than one peak item, simply return one of them.

Input: [1, 5, 3, 2, 4, 0]
Output: 4
Input: [1, 2, 3, 4, 5, 6]
Output: 6
Input: [7, 6, 5, 4, 3, 2]
Output: 7

Describe a divide-and-conquer algorithm that solves this problem in $O(\log n)$ time where n is the size of the array. (***5 marks***)

3. Greedy Algorithms **[10 marks]**

(a) Show how the greedy algorithm for making change gives minimum number of coins for \$7.95 using only New Zealand coinage (200, 100, 50, 20, 10 and 5 cents). Explain your answer. . (***3 marks***)

(b) List and briefly describe the three methods presented in class for proving correctness of greedy algorithms. (***3 marks***)

(c) Consider the following simple greedy algorithm for properly coloring the vertices of a graph. [Recall that a graph is *properly colored* if we can assign colors to vertices such that adjacent vertices have different colors.]

function *GreedyColor*(Graph $G = (V, E)$)
Sort V by their degree in non-increasing order (e.g. largest to smallest)
ColorsAvail = $\{1, 2, \ldots, |V|\}$; ColorsUsed = $\{\}$
for each $v \in V$ (preserving sorted order) **do**
  Let c be smallest positive integer in ColorsAvail not used for a neighbor of v
  Color[v] = c
  ColorsUsed = ColorsUsed $\cup$ $\{c\}$
return Color, |ColorsUsed|

Show with a small counter-example that this greedy algorithm does not always give the minimum number of colors. (Justify your answer by showing a valid smaller proper coloring.) (*4 **marks***)