COMPSCI 320SC 2017 Midterm Test

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 below and just before each answer area.

This 50 minute test is worth 10% of your final grade for the course.

Question #:	1	2	3	4	Total
Possible marks:	6	6	6	6	24
Awarded marks:					

University ID: ______________________________

Student Name: ______________________________

Student Signature: ______________________________

Time Finished: ______________________________

1. For each of the following statements. State whether it is True/False (1 mark) and justify your answer (2 marks).

   (a) $\frac{\lg n}{\sqrt{n}}$ is not $O(\sqrt{n})$. (***3 marks***)

   False.
   In Assignment 2 (Q1b) we showed that $\frac{\lg n}{\sqrt{n}}$ is $O(\sqrt{n})$.

   (b) If $f(n) \in O(n)$ then $f(n)^2 \in O(n^2)$. (***3 marks***)

   True.
   By definition we have constants $n_0 > 0$ and c such that for all $n > n_0$, $f(n) \leq cn$.
   Then, $f(n)^2 \leq (cn)^2 = c^2n^2 = c_1n^2$ where $c_1 = c^2$ for all $n > n_0$.

2. Consider the following Python "divide-and-conquer" function:

```
def excited(L,n):
    if n <= 2: return
    if n == 3: return excited('*'+L+'*',4)
    print(L)
    return excited(L, n//2)
```

Let $T(n)$ denote the number of lines of output generated by a call of excited(L,n).

(a) Provide a recurrence equation for $T(n)$. (***2 marks***)

$$T(n) = \begin{cases} 0 & \text{if } n \leq 2 \\ T(\lfloor n/2 \rfloor) + 1 & \text{otherwise} \end{cases}$$

(b) What are the values of $T(3)$, $T(4)$, $T(5)$, $T(6)$, $T(7)$ and $T(8)$? (***2 marks***)

$T(3) = T(4) = T(5) = 1$ and $T(6) = T(7) = T(8) = 2$

(c) Solve the recurrence exactly for n being a power of 2 (i.e., $n = 2^k$ for $k > 1$). (***2 marks***)

$$T(2^k) = T(2^{k-1}) + 1 = T(2^{k-2}) + 1 + 1 = 0 + \underbrace{1 + 1 + \cdots + 1}_{k-1} = k - 1 = \lg n/2$$

3. (a) State the Red Rule (hint: cycle) and Blue Rule (hint: cutset) and the generic algorithm we used as a basis for proving correctness of several greedy Minimum Spanning Tree (MST) algorithms. (***3 marks***)

Red Rule: Let C be a cycle with no red edges. Select an uncolored edge of C of max weight and color it red.
Blue Rule: Let D be a cutset with no blue edges. Select an uncolored edge in D of min weight and color it blue.
Greedy algorithm: Apply the red and blue rules (non-deterministically!) until all edges are colored. The blue edges form an MST.

(b) Briefly explain Prim, Kruskal and Boruvka's MST algorithms in terms of the Red and Blue Rules. (***3 marks***)

Prim: Start with any node in set S and repeatidly apply blue rule to S and $V \setminus S$.
Kruskal: As progressing through sorted edges: if edge creates cycle then color red; if edge combines tree then color blue.
Boruvka: Apply blue rules (possibly in parallel) until one tree.

4. Below are two Python functions that try to evaluate a polynomial $p(x) = a_0+a_1x+a_2x^2+\cdots+a_nx^n$ of degree n. For each case indicate the exact number of multiplications and additions in terms of n (2 marks) and state True/False whether each correctly evaluates the polynomial at x (1 mark).

(a)
```
def eval1(a,n,x):
    p,xpwr = a[0],x
    for i in range(1,n):
        p += a[i]*xpwr
        xpwr *= x
    return p + a[n]*xpwr
```
(***3 marks***)

True, $2n-1$ multiplications and n additions.

(b)
```
def eval2(a,n,x):
    p = a[n]
    for i in range(n-1,0,-1):
        p = p*x + a[i]
    return p*x + a[0]
```
(***3 marks***)

True, n multiplications and n additions.