COMP 4030/6030 - Spring 2017
Exam 1

Name: Cliff Montjoy

Note: Do not write on the backs; only front pages are digitized and graded.

1. (*20 points*) Answer with True or False.

20

- $3n^2 + n^3 \in \Omega(n^3)$ True
- $3n^2 + n^3 \in O(n^4)$ True

2. (*10 points*) Use the definition of O to show that $5n^2 + 10n \in O(n^2)$.

10

$5n^2+10n \leq 5n^2+10n^2$, ∴ $5n^2+10n \leq 15n^2$ for all $n \geq 1$.
$c=15$. Thus, $5n^2+10n \in O(n^2)$.

3. (*20 points*) Use the definition of Θ to show that $3n^3 + 5n - 5 \in \Theta(n^3)$.

20

$3n^3+5n-5 \leq 3n^3+5n^3-5$; ∴ $3n^3+5n \leq 8n^3$ for all $n \geq 1$.
$c=8$. Thus $3n^3+5n-5 \in O(n^3)$
$3n^3+5n-5 \geq 3n^3$ for all $n>0$; Thus, $3n^3+5n-5 \in \Omega(n^3)$
Since $3n^3+5n-5$ exists in $O(n^3)$ and $\Omega(n^3)$, $3n^3+5n-5 \in \Theta(n^3)$

4. (*10 points*) Use Θ to specify the running time of the following procedure in terms of n.

10

```
sum = 0 constant
for i in range(n): n steps
    for j in range(5): 5 steps (constant)
        sum = sum + i + j
```

$T(n)=\Theta(n)$

5. (*10 points*) Use Θ to specify the running time of the following procedure in terms of n.

10

```
sum = 0   constant
for i in range(n): n
    j = n constant
    while j > 1: logn
        sum = sum + i*j constant
        j = j / 2
```

$T(n)=\Theta(n \log n)$

6. (*10 points*) Is the running time of the procedure in Question 4 is in $\Omega(n)$? Explain briefly.

10

Yes, $\Omega(n)$ specifies a lower bound running time of approximately n steps. n steps will take place regardless of input because of the for loop. Thus, the procedure will take at least n steps.

1

7. (*10 points*) What is the space complexity of the procedure in Question 4? $S(n)=\Theta(n)$

8. (*10 points*) Explain what the following function does. The input is a list of numbers.

10

```
def foo(L):
    if len(L) == 0:
        return 1
    return L[0] * foo(L[1:])
```

If passed an empty list, foo returns 1. Otherwise, it recursively computes the product of numbers in L and returns that product. The recursive call multiplies the first list item by the return value of foo when it is passed the same list, minus the first item.

9. (*10 points*) Use mathematical induction to explain that the following function correctly adds up all numbers that are divisible by 5 in the input list.

```
def bar(L):
    if len(L) == 0:
        return 0
    if L[0] % 5 == 0:
        return L[0] + bar(L[1:])
    return bar(L[1:])
```

Let k equal the number of items in list L. bar works for the smallest case of list length k=0 because there are no numbers divisible by 5 in the list. Let L[1:] be a list of length k-1. Assuming bar works for all cases k-1, bar also returns a value divisible by 5 + the sum of numbers in list of length k-1 which are divisible by 5. Thus bar works for all cases of list length k.

10

10. (*10 points*) The input of the following function is a binary tree T. Assume that such a tree T has 3 attributes: T.left, T.right (both of which are also binary trees), and T.color (which is a string of either "red", "green", or "blue"). If T is empty, the function *is_empty*(T) returns True (if not, it returns False). Complete defining the following Python function so that it correctly counts the number of red nodes in the input binary tree T.

```
def count_red_nodes(T):
    # fill in your codes below to count the number of red nodes in the binary tree T
    if is_empty(T):
        return 0
    count = 0
    if T.color == "red":
        count = 1
    return count + count_red_nodes(T.left) + count_red_nodes(T.right)
```

10