SAN FRANCISCO STATE UNIVERSITY
Computer Science Department

CSC510 Analysis of Algorithms –
Homework 1: Step Counting and Asymptotic Order

[redacted]

Full Name: [redacted] ______________________
Student ID: [redacted] ______________________

Assignment Instructions for students. Must read!

Note: Failure to follow the following instructions in detail will impact your grade negatively.

1. This homework is worth 15 points of your final grade in this course
2. Only homework assignments uploaded to Canvas in (one document) PDF format will be graded.
3. Handwriting work is allowed as long as the work is clear and readable. If we can't understand your work, then we can't grade it.
4. Blank assignments submitted after the due date will be graded as such. It is the responsibility of students to check if the assignment was submitted using the correct format.
5. Late homework submissions will be subject to penalties. A 10% deduction will be applied for each day that the assignment is overdue, up to a maximum of three days. After three days, the assignment will be considered as not submitted and will be graded as such
6. All the material found in this homework was (or will be) covered during lectures. Students are responsible for what they miss in class. Attendance to lectures is essential in order to complete your homework assignments.

Solutions (your work starts here)

1. (3 points) Compute the following summation. Show all your work to get credit

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

$$\sum_{i=10}^{n} \sum_{j=1}^{\log_2 n+1} j$$

$$\sum_{j=1}^{\text{Log}_2 n+1} j$$

$$= \frac{\text{Log}_2 n+1((\text{Log}_2 n+1)+1)}{2}$$

$$= \frac{\text{Log}_2 n+1(\text{Log}_2 n+2)}{2}$$

$$= \frac{\text{Log}_2 n^2+2}{2}$$

$$\sum_{i=10}^{n} 1 \rightarrow \sum_{i=1}^{n-9} 1$$

$$\sum_{i=10}^{n} \sum_{j=1}^{\text{Log}_2 n+1} j = (n-9) \frac{\text{Log}_2 n+1(\text{Log}_2 n+2)}{2}$$

2. (3 points) Compute the number of operations or basic steps executed for the following code as a function of n. Assume for this problem that all the operations found in this code are considered basic operations. **Show all your work to get credit**

```
          x = 0; → 1 operation
          for (i=1; i<=n; i=i*3) { →
Log3 n    ← print("1")
Log3 n-1  ← if (x > 0) {
Log3n     ←     print("2")
                for (j=1; j<=2n; j++) {
                    print("3")
                }
            }
            for (k=6; k<=n; k++) {
1 operation ←   print("4")
            }
1 operation ← print("5")
1 operation ← x++;
          }
```

This for Loop runs as Long as i is Less than or equal to n, i is multiplied by 3 each time. i= 1 (1 operation) and goes all the way to n. $i <= n \rightarrow \text{Log}_3 n + 1$

$i = i * 3 \rightarrow \text{Log}_3 n$

for (j = 1 ; j <= 2n ; j++)

$\uparrow$ 1 operation $\quad \uparrow$ 2n $\quad \uparrow$ n $\quad = 2n$

for (k = 6 ; k <= n ; k++)

$\uparrow$ 1 operation $\quad \uparrow$ n - 6 + 1 = n - 5 $\quad \uparrow$ n

$$\Rightarrow 1 + \text{Log}_3 n + 1 + \text{Log}\, n - 1 + 2n + n - 5 + 1 + 1$$

$$= 3 + 2\text{Log}_3 n + 2n + n - 5$$

$$t(n) = 2\text{Log}_3 n + 2n + n - 2$$

3. (3 points) Compute the big O time complexity (worst case) of the following code. Assume that for this problem that the print statement is the basic operation. **Show all your work to get credit, including a table of steps to n (as seen in class), and then all the summations and all your work to compute the time complexity.**

$O(n) \leftarrow$ `for (i=1; i<=(n-2); i++) {`

$O(n) \leftarrow$ `for (j=i+1; j<=(n-1); j++) {`

$O(n) \leftarrow$ `for (k=j+1; k<=n; k++) {`

```
for (i=1; i<=(n-2); i++) {
    for (j=i+1; j<=(n-1); j++) {
        for (k=j+1; k<=n; k++) {
            print("Hello Class");
        }
    }

}
```

i	j	k	Print statement
1	2	3	1
1	2	4	1
1	2	5	1
⋮	⋮	⋮	⋮
1	2	n	1
1	3	4	1
1	3	5	1
⋮	⋮	⋮	⋮
1	3	n	1
n-2	n-1	n	1

$$O(n \times n \times n) = O(n^3)$$

$$t(n) = \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} 1$$

$$\Rightarrow \sum_{k=j+1}^{n} 1 = n-(j+1)+1 = n-j$$

$$\Rightarrow \sum_{j=j+1}^{n-1} n-i = \sum_{j=i+1}^{n-1} n - \sum_{j=i+1}^{n-1} i = n(n-i-1) - (i*(n-i-1))$$

$$\Rightarrow \sum_{i=1}^{n-2} (n(n-i-1)) - (i(n-i-1))$$

$$= (n-2)(n-1)(n-3) + (n-3)(n-2)(n-4) + \dots + 2(1)(0)$$

$$= O(n^3)$$

4. (3 points) Using sequential search:

The probability that the item **IS NOT** in the array is $\frac{1}{4}$. If the item is in the array, the probability that the last item in the array matches the search key is $\frac{1}{8}$. The probability that the next to the last item of the array matches the search key is $\frac{1}{5}$. The probabilities matching any of the remaining items are all equal.

1. (1 point) What is the probability (as a function of n) of matching one of the 1st through $(n-2)nd$ items? **Show all your work to get credit**

$$P_{out} = \frac{1}{4} \; ; \; P_{in} \left(P_n \left(\frac{1}{8}\right)\left(\frac{3}{4}\right) + P_{n-1}\left(\frac{1}{5}\right)\left(\frac{3}{4}\right) + P_{1 \ldots n-2}\left(\quad \right)\right) = \frac{3}{4}$$

$$P_{1 \ldots n-2} = P_n\left(\frac{1}{8}\right)\left(\frac{3}{4}\right) + P_{n-1}\left(\frac{1}{5}\right)\left(\frac{3}{4}\right)$$

$$P_{1 \ldots (n-2)} = \frac{3n}{32} + \frac{3n-3}{20} = \frac{15n}{160} + \frac{24n-24}{160}$$

$$= \frac{39n - 24}{160}$$

2. (1.5 points) Assuming that comparison of an array item where the search key is the basic operation, what is the average case complexity function for sequential search under these conditions. **Show all your work to get credit**

$$A(n) \quad \frac{n}{4} + \frac{3n}{32} + \frac{3n-1}{20} + \frac{1}{160(n-2)} \sum_{i=1}^{n-2} i$$

$$= \frac{40n + 15n + 8n - 8}{160} + \frac{1}{160\cancel{(n-2)}}\left[\frac{\cancel{(n-2)}(n-1)}{2}\right]$$

$$= \frac{63n - 8}{160} + \frac{(n-1)}{320} = \frac{127n - 17}{320}$$

3. (0.5 points) What is the big O average case time complexity under these conditions.

The average case time complexity under these conditions is $O(n)$

5. (3 points) Use the limit ratio approach to determine if the following asymptotic orders are true or false. **Show all your work in detail to get credit for this problem including your limit ratio computations**

$\Omega(g(n)) \rightarrow f(n) \geq g(n)$

$\rightarrow \frac{f(n)}{g(n)} \rightarrow > 0$

$$2^n + \underline{n^n} \in \Omega(\underline{n!})$$

$$\lim \frac{f(n)}{g(n)} = \frac{n^n}{n!} = \infty$$

So here is $\boxed{\text{true}}$

$\theta(g(n))$ $f(n) = g(n)$

$$16^{\log_4 n+2} + \sqrt{n} \in \Theta(n^2)$$

$$\lim_{n\rightarrow} \frac{f(n)}{g(n)} = \frac{16^{\log_4 n+2}}{n^2} =$$

$$16^{\log_4 n} = 2^{4 \log_2 2n} = 2^{4 \frac{1}{2} \log_2 n}$$
$$= n^2$$

$$\lim \frac{f(n)}{g(n)} \Rightarrow \frac{n^2}{n^2} = 1 \Rightarrow \boxed{\text{True}}$$

$\omega(g(n)) \rightarrow f(n) > g(n)$

$\frac{f(n)}{g(n)} \rightarrow \infty$

$$\underline{\frac{\log_2(\log_2 n)}{(\log_2 n)^{-1}}} + 100 \in \omega(\underline{\log_2 n})$$

$$\lim \frac{f(n)}{g(n)} = \frac{\frac{\log_2(\log_2 n)}{(\log_2 n)^{-1}}}{\log_2 n} = \frac{\log_2(\log_2 n)(\log_2 n)}{\log_2 n} = \log_2(\log_2 n)$$
$$= \infty$$

$\boxed{\text{So True}}$

$$\underline{n^4} + n^3 + 10000 \in o(\underline{n^4})$$

$o(g(n)) \rightarrow f(n) < g(n)$

$\rightarrow \frac{f(n)}{g(n)} \rightarrow 0$

$$\lim \frac{f(n)}{g(n)} = \frac{\cancel{n^4}}{\cancel{n^4}} = 1$$

So this is $\boxed{\text{false}}$