

CS210 Data Structures and Algorithms Fall 2024

Assignment #2

TOTAL POINTS: 25

Name:

Kefy

1. What is the tightest bound Big O for the following: (10 Points)

a) $f(N) = (1/2)(N \log N) + (\log N)^2$

$n \log n$

b) $f(N) = N^2 * (N + N \log N + 1000)$

$n^3 \log n$

c) $f(N) = N^2 \log N + 2^N$

2^n

d) $f(N) = ((1/2)(3N + 5 + N))^4$

n^4

e) $f(N) = (2N + 5 + N^4) / N$

n^3

f) $f(N) = \log_{10}(2^N)$

n

g) $f(N) = N! + 2N$

$n!$

h) $f(N) = (N * N * N * N + 2N)^2$

n^8

i) $f(N) = N^{1/2} + \log N$

either is correct
 $\text{they switch in order}$

j) $f(N) = N \log (100^3)$

n

2. Give the tightest bound in terms of Big O, of the following code snippets. It is not required to show your work. Only mention Big O. (6 points)

a. `public type something(n) {
 result = 0;
 while (n > 1){
 n /= 2;
 result += 1;
 }
 return result;
}`

$O(\log n)$

b. `public type something (n, a[]){
 for (i=0, i<n*n; i++){
 if (a[i] == 0)
 return 0;
 }
 return 1;
}`

$O(n^2)$

c. `public type something(n) {
 result = 0;
 for (i=0, i<n; i*=3){
 for (int j=i; j<n;j++){
 result += 1;
 }
 }
 return result;
}`

~~$O(n \log n)$~~

infinite loops do not have Big O

Multiple Choice Questions (4 Points)

3. Arrange the following in increasing order of asymptotic complexity

$$F1(n) = n^2 + 10n \quad n^2$$

$$F2(n) = \log n + 89 \quad \log n$$

$$F3(n) = n \log n + 2^n \quad 2^n$$

a. F1, f2, f3	b. F2, f1, f3
c. F2, f3, f1	

4. What is the runtime efficiency of the following code

for (i=0; i<n-3; i++)

$$a[i] = 0$$

a. O(n)	b. O(n ²)
c. O (log n)	d. O (1)

5. Which assumption is not required for the RAM model of computation?

a. unit time per data storage or retrieval	b. infinite memory
c. unit time per mathematical operation	d. presorted data

6. Why is the following instruction not an algorithm?

If you see a bear, run.

a. It does not say which way to run.	b. It does not specify the kind of bear.
c. Neither the problem nor the solution steps are precisely defined.	d. There is no looping or repeat instruction.

Short Answer (5 Points)

7. For the code snippet given below. Calculate (using summation), the total number of times the inner most statement in the for loop is executed.

What kind of summation is this? Expand the summation to find the Big O of the code snippet to the tightest upper bound.

```
for (int i = 0; i < n; i++) {
    for (int j = 0; j < pow(3, i); j++) {
        total_iterations++;
    }
}
```

Outer loop runs from $i=0$ to $i=n-1$
 $O(n)$ times.

Inner loop runs $j=0$ to $j=3^i-1$ for each i .

Total number of loop runs = $3^0 + 3^1 + 3^2 + \dots + 3^{n-1}$

$$\Rightarrow \sum_{i=0}^{n-1} 3^i$$

This is a geometric series

$$\sum_{i=0}^{n-1} r^i = \frac{r^n - 1}{r - 1}$$

$$\sum_{i=0}^{n-1} 3^i = \frac{3^n - 1}{3 - 1} = \frac{3^n - 1}{2} \in O(3^n)$$