

CSE 214 Spring 2023 (Student Version)

Recitation 1 – Order of Complexity

1. [5 minutes] Order the following from most to least efficient:

$\{O(n^2), O(n \log n), O(n^n), O(10n), O(n!), O(1), O(n^{10}), O(\log n), O(n^3), O(10^n)\}$

2. [10 minutes] What is the time complexity of the following algorithms:

- A program that first fills an array of size n with random integers, then sorts them using an $O(n^2)$ algorithm.
- An $O(n \log n)$ algorithm that sorts an array of n integers from smallest to largest and then returns the smallest integer.
- Using binary search to find a target element in an integer array.
- Retrieving the i^{th} element from an array of size n .

3. [5 Minutes] What is the order of complexity for the following code:

```
int i, j, k, x = 0;
for (i = 0; i < n; i += 2) {
    for (j = 2; j < n; j *= 2) {
        x = x + n / 2;
    }
}
for (k = 0, k < n, k++) {
    x++;
}
```

4. [5 minutes] Why is it that we use $O(\log n)$ without specifying the base of the log? (Ex: $O(\log_2 n)$)

5. [5 minutes] Write the order of complexity for the following programs using Big O notation if the programs execute the following number of operations for n inputs:

a. 66

b. $n - n^3 + 7n^5$

c. $\frac{1}{2}n^2 + 1000\log(n) + \log(n^n)$

d. $9^n + n! + 3n^{45n}$

e. $n^{\frac{1}{37}} + \log(n)$

f. $O(10^n) + O(43^n) + O(85^n) + O(n^{99})$

6. [10 min] Suppose we have a list of 1,024 elements. Searching for a value in a list:

a) Assume the list is not sorted. To perform binary search, we must first sort the list. We can use a sorting algorithm that has a complexity of $n \log(n)$. If it takes 200 ms to complete the sorting process, how long will it take to sort a list of 262,144 elements using the same algorithm?

b) Assume now the list is currently sorted. Using binary search, we can find our target value using an algorithm with complexity of $\log(n)$. If using this algorithm takes 50 ms to complete, how long will it take if we increased the list size to 1,048,576?

7. [10 minutes] Code Analysis: What is the order of complexity of the following blocks of code?

a)

```
int i = n, j, k = 0;
while(i > 0){
    for(j = 0; j < i; j++){
        k++;
    }
    i /= 3;
}
```

b)

```
int i, j, k= 0;
for (i = n; i > 0; i /= 4) {
    for (j = 0; j < n; j+=2) {
        ++k;
    }
}
```