

CS 1331 – Big-O, Searching, and Sorting Worksheet

NOTE: THIS IS NOT A PRACTICE EXAM: It is not meant to in any way reflect the contents or format of Exam 4. This is a practice worksheet and is not meant to be the sole preparation for the exam. Questions on this worksheet are meant to give students a better understanding of course concepts for homeworks as well as future exams.

Big-O

```
public int mysteryFunction(int[] arr) {  
    int halfSum = 0;  
    for (int i = 0; i < arr.length; i+=2) {  
        halfSum += arr[i];  
    }  
}
```

1. What is the Big-O of mysteryFunction?

```
public int bigMystery(int[] arr) {  
    for (int j = arr.length - 1; j > 0; j / 2) {  
        mysteryFunction(int[] arr);  
        //arr[j] = arr[j - 1];  
    }  
}
```

2. What is the Big-O of bigMystery?
3. If bigMystery used the commented code instead of calling mysteryFunction, what would be it's new Big-O?

```

public int newFunction(ArrayList<Double> alist) {
    int n = alist.size() * 2;

    int j = 1000000000000;

    int bigNum = 0;
    for (int i = 0; i < j; i++) {
        ++bigNum *= 2;
    }
}

```

4. What is the Big-O of newFunction?
5. If n replaced j in the for loop, what is the new Big-O?

Searching

1. Given the array [2, 3, 6, 7, 8, 11, 20, 23, 25], write out the subarrays we will look at in a Binary Search to find 6. Indicate which element is the middle index for each iteration.
2. Given the array [4, 13, 23, 44, 56, 67, 68, 72, 73, 80], write out the subarrays we will look at in a Binary Search to find 74. Indicate which element is the middle index for each iteration.
3. Which searching (Linear and Binary) algorithms can you use on the given array?
Why?
[4, 3, 7, 2, 6, 8]

Sorting

Bubble Sort: given the unsorted array, how many iterations it would take to completely sort the array? Write out how the array looks after each complete iteration.

Given Array: [5, 8, 2, 4, 3, 9, 1, 10, 0]

Selection Sort: given the unsorted array, select the answer that shows how the array will look after 4 iterations of selection sort.

Given Array: [45, 67, 24, 89, 34, 44, 20, 90, 4]

- A) [4, 20, 24, 34, 89, 44, 67, 90, 45]
- B) [4, 20, 24, 34, 44, 89, 67, 90, 45]
- C) [4, 24, 34, 67, 89, 44, 20, 90, 45]
- D) [4, 20, 24, 34, 44, 45, 67, 89, 90]

Insertion Sort:

1. True or False: After $k + 1$ iterations of insertion sort, the first k elements are sorted but not in their final positions
2. Given the unsorted array, how many iterations would it take to sort the array? Write out how the array looks after each complete iteration.

Given Array: ["w", "r", "h", "u", "i", "b"]

Note: "a".compareTo("b") < 0