//Liang Fang

//Emplid: 23134567

//MAC286

//Take home exam

Question 4:

//Write a program to get 10000 random elements between 1 and 100. Print all the elements in ascending order and descending order. Also discuss the logic of your program and its complexity.

**public** **class** SortingNumbers {

**private** **int**[] arr;

**private** **int** nElems;

**public** SortingNumbers(**int**[] arr){

**this**.arr = arr;

nElems = arr.length;

}

**public** **void** ascendingBubbleSort(){

**int** out, in;

**for**(out = nElems - 1; out>1; out --){

**for**(in=0; in<out; in++){

**if**(arr[in] > arr[in+1])

swap(in, in+1);

}

}

}

**public** **void** descendingBubbleSort(){

**int** out, in;

**for**(out = nElems - 1; out>1; out --){

**for**(in=0; in<out; in++){

**if**(arr[in] < arr[in+1])

swap(in, in+1);

}

}

}

**public** **void** swap(**int** a, **int** b){

**int** temp = arr[a];

arr[a] = arr[b];

arr[b] = temp;

}

**public** **void** display(){

**for**(**int** i=0; i<arr.length;i++){

System.***out***.print(arr[i] + " ");

}

System.***out***.println("");

}

}

**import** java.util.\*;

**public** **class** SortingNumbersApp {

**public** **static** **void** main(String[] args) {

**int** size = 10000;

**int**[] arr = **new** **int**[size];

**for**(**int** i =0; i< size; i++) {

arr[i] = (**int**)(Math.*random*()\*100) +1;

}

SortingNumbers sn = **new** SortingNumbers(arr);

System.***out***.println("The 10000 numbers are: ");

sn.display();

System.***out***.println("The ascending order is: ");

sn.ascendingBubbleSort();

sn.display();

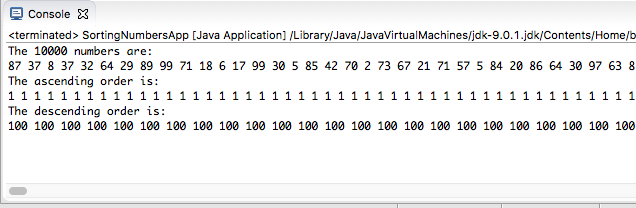
System.***out***.println("The descending order is: ");

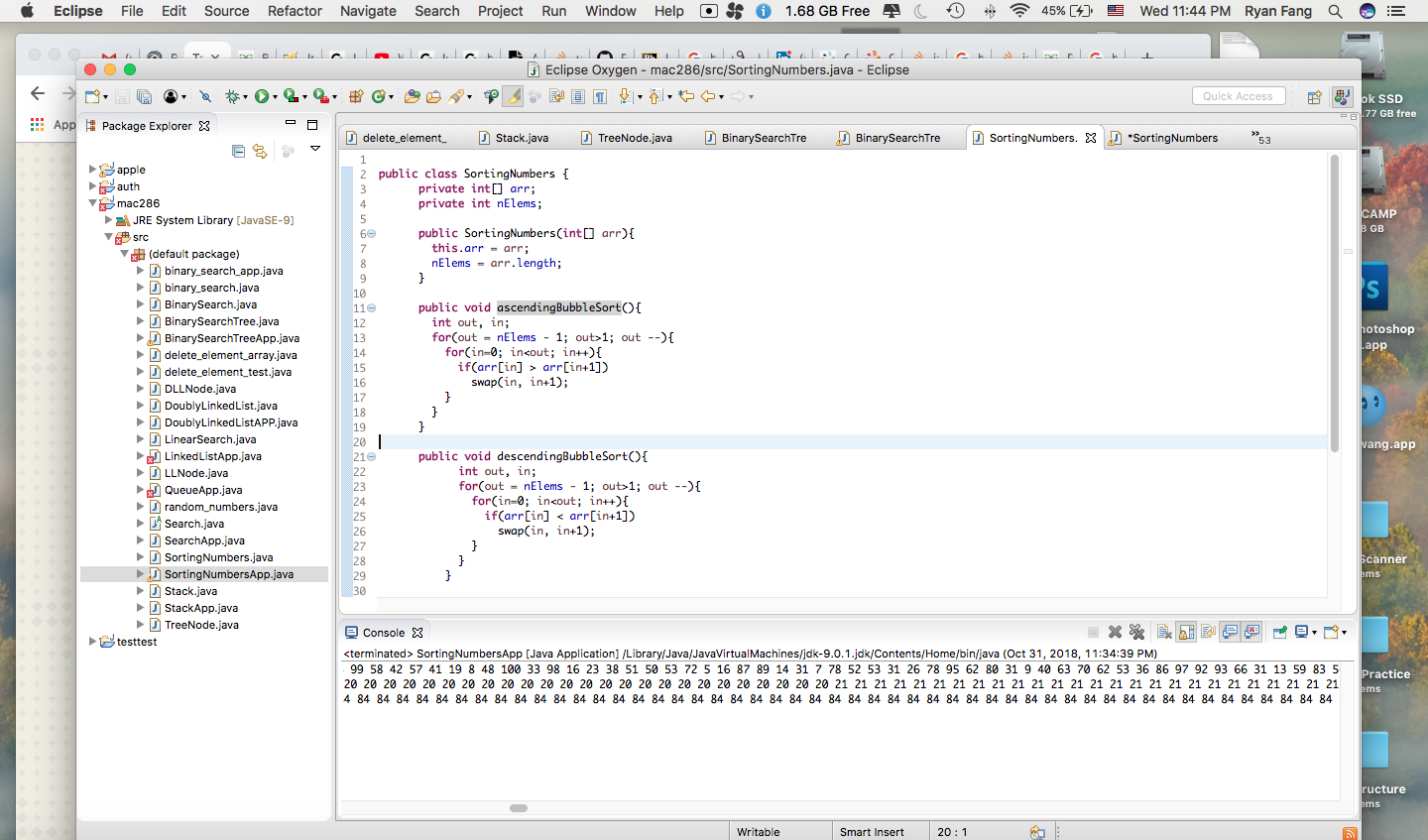
sn.descendingBubbleSort();

sn.display();

}

}





**Analysis:**

I used bubble sort to sort those 10000 random numbers. Generally ,the bubble sort is not efficient to sort numbers because it takes O(n^2) time in terms of time complexity. For space complexity, it only takes O(1) because it did not take extra memory. Space is only used for a temp variable for swapping.