CSCI E-22: Paul Zeng

Part I, 1.a



Part I, 1.b Indicate what will be printed by the final line of code shown above

Answer: the final line will print: 12 10 10

Part I, 2.a

public static void swapPairs(int[] arr){

if (arr == null ){

throw new IllegalArgumentException();

}

for (int i = 0; i < arr.length-1; i += 2){

int temp = arr[i];

arr[i] = arr[i+1];

arr[i+1] = temp;

}

}

Part I, 2.b

public static int longestSorted(int[] arr){

if (arr == null) {

throw new IllegalArgumentException();

}

if (arr.length == 0) return 0;

if (arr.length == 1) return 1;

int length = 1; // the length of the longest increasing sequence

int curr\_len = 1; // the length of currently running sequence

// at this point, arr should have at least two elements

for (int i = 0; i < arr.length-1; i++){

if (arr[i] < arr[i+1]){

curr\_len++;

} else {

curr\_len = 1;

}

if (curr\_len > length) {

length = curr\_len; // update the length of the longest increasing sequence

}

}

return length;

}

Part I, 3.a

main calls mystery(10, 1)

mystery(10, 1) calls mystery(7, 2)

mystery(7, 2) calls mystery(4, 3)

mystery(4, 3) calls mystery(1, 4)

mystery(1, 4) returns 1

mystery(4, 3) returns 4

mystery(7, 2) returns 6

mystery(10, 1) returns 7

main()

Part I, 3.b

mystery(10, 1) returns 7.

Part I, 3.c

When the base case is reached, there were 5 frames on the stack

Part I, 3.d

No, because either a <= b, in which case we've already reached the base case, or we go to the recursive case and

a is reduced by 3, and b is increased by 1. Because a is strictly decreasing and b is strictly increasing, there will

be a finite number of recursive calls when the base case is reached.

Part I, 4

public static void printReverse(Object[] arr, int i){

if (i >= arr.length-1){

System.out.println(arr[i]);

return;

} else {

printReverse(arr, i+1);

System.out.println(arr[i]);

return;

}

}