

CSC 212: Data Structures and Abstractions
Spring 2019
University of Rhode Island
Weekly Problem Set #4

Due Thursday 3/7 at the beginning of class. Please turn in neat, and organized, answers hand-written on standard-sized paper **without any fringe**. At the top of each sheet you hand in, please write your name, and ID.

1. Write a recursive function that sums all of the elements of a given n length array, matching this signature: `int sum(int* arr, int n);`
2. Rewrite the recursive sum function to only sum odd numbers within the array.
3. Write a recursive function that can find the minimum of a given array, matching this signature: `int min_array(int* arr, int n);`
4. Reverse the elements of an array in place. Matching the following function signature: `void reverse_array(int* arr, int n);`
5. Write a function to print triangles to `std::cout` that takes three positive integers: a , b , c as input. The function should print the `+` character a times, then $a + c$ times, then $a + c + c$ times, and so on. This pattern should repeat until the line is b characters long. At that point, the pattern is repeated backwards. For example calling `draw_triangle(4, 7, 1)` will output:

```
++++
+++++
++++++
+++++++
+++++++
+++++++
+++++++
+++++
++++
++++
```

6. For both insertion and selection sort, describe if the algorithm is stable and if not give an example array that shows the unstable behavior.
7. Draw out a recursion tree for the functions on the following page using the given inputs: `fib(5)`
`gcd(12,56)`

```
int fib(int n) {  
    if (n <= 1) return n;  
    return fib(n-1) + fib(n-2);  
}
```

```
int gcd(int a, int b) {  
    if(b == 0) return a;  
    else {  
        return gcd(b, a % b);  
    }  
}
```