

In this home work you will implement and study some sorting algorithms.

1. Implement a function mergeSort and another function quickSort to sort a set of integers in increasing order.
2. Plot the running times of the two algorithms against the *size* of the input when the input arrays are randomly sorted. Do the same when the inputs are already sorted. What functions best characterize the running times? Compare your results with what is expected from the theoretical analysis. Discuss the results and their implication.
3. The following code implements Bubble Sort which sorts the array a in ascending order. Fill in the assertions which must hold at each of the points indicated below.

```
def bubbleSort(a):  
    swap,j = 1, len(a)  
    while swap == 1:
```

```
# Assertion 1:
```

```
    swap,j = 0,j-1  
    for i in range(j):
```

```
# Assertion 2:
```

```
        if a[i] > a[i+1]:  
            a[i], a[i+1], swap = a[i+1], a[i], 1
```

```
# Assertion 3: [after the while-loop ends].
```

Argue informally that Assertions 1 and 2 hold during the execution and Assertion 3 holds at the end.

4. Express the running time of the above algorithm in  $O$ -notation. Justify your answer.

### How to submit?

- Label the source file 'hw1.py' and the report report1.pdf.
- Upload report1.pdf on the Canvas site before the deadline.
- Upload the source hw1.py also before the deadline at COE's teach interface for Homework1 at <https://teach.engr.oregonstate.edu/> [Links to an external site.](#)
- Please test your program thoroughly before you submit.

### Rubric

- Programs run correctly: 4 points.
- Results and analysis: 2 point
- Bubble Sort analysis: 3 points

- Complexity analysis and justification: 1 point