1.T(n) from high to low
O(n!) 6. 0.0001n!
O(k^n) 5. 3.2^n
O(n^3) 9. $1n^3$ + 5n -100
O(n^2) 8. $5n^2$ + 10n 3. $4n^2$
O($n^{3/2}$) 1. $n^{3/2}$
O($n * logn$) 7. n*$\log_2$n
O(n) 2. 100n
O($\sqrt{n}$) 10. $\sqrt{n}$
O(logn) 4.$\log_2$n

2.

```
# SEGMENT 1
a_len = len(n)
if a_len == 1: # TRIVIAL CASE WHERE N IS A SINGLE VALUE
        return n[0], n[0]
```

O(1)

```
# SEGMENT 2 a = merge_sort(n) # USE WHAT YOU KNOW ABOUT TIME COMPLEXITY OF MERGE SORT FOR THIS SEGMENT
```

O(n*logn)

```
# SEGMENT 3
split = a_len // 2
median: int = 0
if a_len % 2 == 0: # is even
        median = (a[split - 1] + a[split]) / 2
else: median = a[split]
```

O(1)

```
# SEGMENT 4
running_sum = 0
for i in range(0, a_len):
        running_sum += a[i]
mean = running_sum / a_len
return (median, mean)
```

O(n)

T(n) = O(1) + O(n*logn) + O(1) + O(n) = O(n*logn)