```
fun printFirst100(ns: List<Int>) {
    for (i in 0..100)
        println(ns[i])
```

treeSet += "a new element"

# Time Complexities Game





O(log n)























list.sorted() has a medium time complexity of O(n log n) why is QuickSort being used and not a O(n log n) sorting algorithm?

### list.sorted()

## Time Complexities Game

```
O(1)
                                                                     O(log n)
fun printFirst100(ns: List<Int>) {
                                         treeSet += "a new element"
    for (i in 0..100)
        println(ns[i])
O(n)
                                                                      O(n^2)
                                         list.sorted()
```

list.sorted() has a medium time complexity of O(n log n) why is QuickSort being used and not a O(n log n) sorting algorithm?

### Time Complexities Game

```
O(1)
                                                                     O(log n)
fun printFirst100(ns: List<Int>) {
                                         treeSet += "a new element"
    for (i in 0..100)
        println(ns[i])
O(n)
                                                                      O(n^2)
                                         list.sorted()
```

list.sorted() has a medium time complexity of O(n log n) why is QuickSort being used and not a O(n log n) sorting algorithm?