CS 5012: Foundations of Computer Science

UNIVERSITY OF VIRGINIA
DATA SCIENCE INSTITUTE

Asymptotic Complexity Exercise – SOLUTIONS

▶
```
def measure(inputList):
    int n = len(inputList)
    int sum = 0;
    for i in range(0, n):
        for j in range(0, 5):
              sum += j * inputList[i]
       for k in range(0, n):
           sum -= inputList[k]
```

The asymptotic complexity of this algorithm is: O (________)
SOLUTION: O (n^2) *Quadratic time*

▶
```
def addElement(ele):
    myList =   [ ]
    myList.append(666)
    print myList
```

The asymptotic complexity of this algorithm is: O (__________)
SOLUTION: O (1) *Constant time*

▶
```
num = 10

def addOnesToTestList(num):
    testList = []
    for i in range(0, num):
        testList.append(1)
        print(testList)
    return testList
```

The asymptotic complexity of this algorithm is: O (________)
SOLUTION: O (n) *Linear time*

▶
```
testList = [1, 43, 31, 21, 6, 96, 48, 13, 25, 5]

def someMethod(testList):
   for i in range(len(testList)):
       for j in range(i+1, len(testList)):
          if testList[j] < testList[i]:
            testList[j], testList[i] = testList[i], testList[j]
          print(testList)
```

The asymptotic complexity of this algorithm is: O (________)
SOLUTION: O (n^2) *Quadratic time*

▶
```
def searchTarget(target_word):
   for (i in range1):
       for (j in range2):
            for (k in range3):
              if (aList[k] == target_word):
                  return 1
       return -1
   return -1
```

The asymptotic complexity of this algorithm is: O (________)
SOLUTION: O (1) *Constant time* *(rangeX lists are not part of the problem size, target_word)*
Also accepted: O (a*b) *a is size of range2 and b size of range3 (both a and b are constants)*

▶
```
def someSearch(sortedList, target):
    left = 0 right = len(sortedList) - 1
    while (left <= right):
       mid = (left + right)/2
       if (sortedList(mid) == target):
          return mid
       elif(sortedList(mid) < target):
          left = mid + 1
       else:
          right = mid - 1

    return -1
```

The asymptotic complexity of this algorithm is: O (________)
SOLUTION: O (log n) *Logarithmic time*