CS 5012: Foundations of Computer Science

Asymptotic Complexity Exercise

Given the following code snippets, provide the time complexity in the form of Big-O notation. Justify your response and state any assumptions made. Treat these functions as constant runtime: print(), append()

```
► 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 (_____)

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

The asymptotic complexity of this algorithm is: O (_____)

```
► 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 (_____)

```
► 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 (_____)

```
► def searchTarget(target_word):
► # Assume range variables are unrelated to size of aList

    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 (_____)

```
► 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 (_____)