CPSC 319 Tutorial 01
Finding Asymptotic Complexity: Examples

Longsheng Zhou

Department of Computer Science
University of Calgary

January 20, 2015

Objectives

Jan.20	Examples for Introduction.
Jan.22	Group Exercises.

Some Info

1. Group Exercises
2. About programming language: Java
3. Any other questions?

Supplementary Instruction (this part was not presented in tutorial):

"In most cases, we are interested in time complexity, which usually measures the number of assignments and comparisons performed during the execution of a program. Chapter 9, which deals with sorting algorithms, considers both types of operations; ***this chapter considers only the number of assignment statements."***

–Testbook "Data Structure and Algorithms in Java" page 64.

This words tell us what kind of operations should be counted in the (time) complexity analysis. Thus, the slides used in today's tutorial are modified. Now, we only count the assignment statements in our analysis. To sum up, ***println*** statement, $i - 1$, etc are not counted anymore, but *sum+=a[j], i++,* etc assignment statements are still counted.

Example

```
/*Code 1*/
for(i=sum=0; i < n; i++)
      sum+ = a[i];
```

Complexity Analysis:

- two variables, i.e. i, sum are initialized (**2** assignment operations).
- then the for loop iterates n times, because i iterates from 0 to $n-1$.
- during each iteration, it executes **2** assignments, i.e. $sum+=a[i]$ and $i{+}{+}$

Thus the complexity is

$$2+2n=O(n)$$

.

Example

```
/*Code 2*/
for(i = 0; i < n; i++){
    for(j = 1, sum = a[0]; j <= i; j++);
        sum+ = a[j];
    system.out.println(“Sum from 0 through”+i+”is”+sum);
}
```

Complexity Analysis:

- i is initialized, this is 1 assignment operation.
- outer loop is performed n times,because i is iterates from 0 to $n-1$.
- For each i, there are **3** assignment statements, i.e. $j=1$, $sum = a[0]$ and i++. and inner loop is executed i times, because j is iterated from 1 to i. For each j, there are **2** assignment operations, i.e. $sum+=a[j]$ and j++,

ps: here the ”println” statements are not counted anymore based on the textbook assumption. Thus, the complexity is

$$1+3n+2(1+2+...+n-1)=O(n)+O(n^2)=O(n^2)$$

Example

```
/*Code 3*/
for($i = 4; i < n; i++$){
   for($j = i - 3$, sum=$a[i - 4]$; $j <= i$; $j$++)
      sum+=a[j];
   System.out.println("Sum for subarray"+
         (i-4)+"through"+i+"is"+sum);
}
```

Complexity Analysis:

- i is initialized by 4, 1 assignment operation..
- outer loop is performed $n - 4$ times,
- for each i, there are **3** assignment statements, i.e.$j = i-3$, $sum{=}a[i-4]$ and $i{+}{+}$. and the inner loop is executed 4 times for each i, and each time it performs **2** assignment operations, i.e. $sum+ = a[j]$ and $j{+}{+}$

Thus the complexity is

$$1 + (n - 4)(3 + 4 * 2) = O(n)$$

Thank you!

AUTHOR: Longsheng Zhou
ADDRESS: ICT 609e
Department of Computer Science
University of Calgary
EMAIL: lozhou@ucalgary.ca