

CS367 Lecture 5

Monday 23 June 2014

Class Announcements/Reminders:

- HW1 due 11 PM tonight. No late submissions accepted.
- P1 extension by one day, minor corrections to documentation and code.
- ODT files for outlines
- Eclipse, Java resources on class homepage

Last class:

- Primitives vs. References (finish)
- Command-Line Java
- Exceptions

Today:

- Exceptions – Behind the Scenes
- Complexity
 - Concepts, Definitions, Assumptions, Examples
 - Growth of functions and notation

Exceptions – Behind the Scenes

Normal execution mode

Exception-handling mode

- If in a `try` block

- If not in a `try` block

Back to normal execution mode – but where?

Complexity - Intro

Quantifying “efficiency”

Examples

Resources: number of “basic” operations (time) and memory used (space)

Model of computation

Complexity: Predicting how resources required grow with problem size

Complexity: Examples

Number-guessing game

Array assignment loop:

```
for (int i = 0; i < n; i++) {  
    a[i] = i;  
}
```

Complexity: Insertion sort

Pseudocode:

```
INSERTION-SORT(A)
1   for (j = 1 to A.length - 1)
2       key = A[j]
3       i = j - 1
4       while (i >= 0 and A[i] > key)
5           A[i + 1] = A[i]
6           i = i - 1
7       A[i + 1] = key
```

Best-case running time

Worst-case running time

Complexity: Simplifying Assumptions

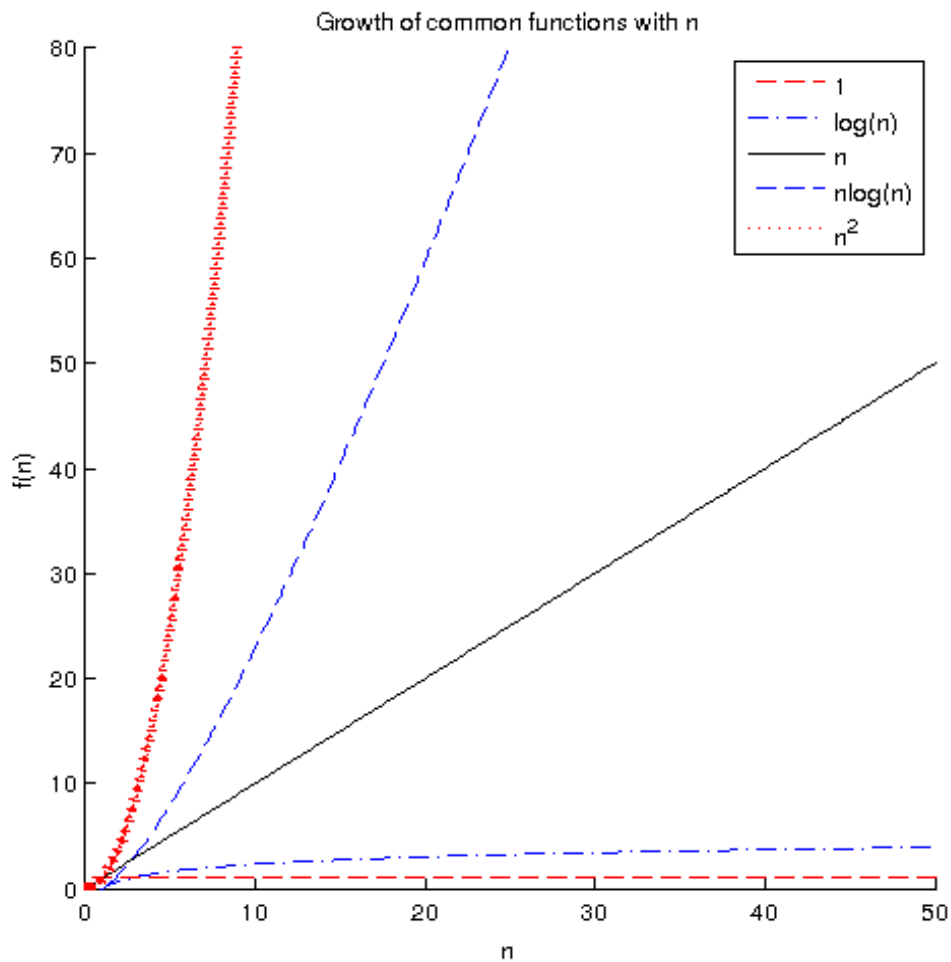
Complexity: How functions grow

Higher order terms compared to rest

Example functions:

Complexity: How functions grow (cont'd)

n	$n \log(n)$	n^2	2^n	$n!$
2	2	2	4	2
4	8	16	16	24
10	33.2	100	1024	3628800
20	86.4	400	1048576	
100	664.4	10000		
1000	9965.8	1000000		



Asymptotic efficiency: Theta and Big-O notation

Definitions:

Example:

Abusing notation