CS367 Announcements
Tuesday, June 25, 2013

- H2 released

Last Time

- Exceptions
- Complexity

Today

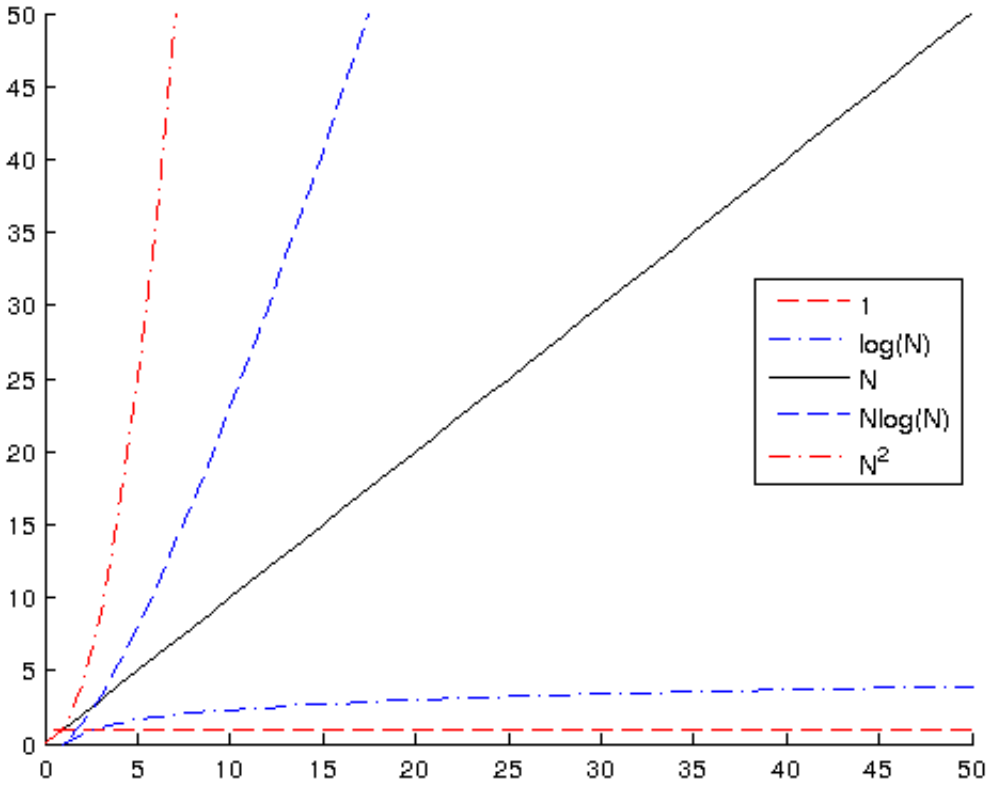
- Complexity (cont.)

Analyzing Algorithm Efficiency

complexity =

If problem size doubles and the # of operations:

Comparing 1, $\log(N)$, N, $N\log(N)$, N^2



Example: Giving a Toast

Number Guessing Game

Person 1 picks a number between 1 and N

Repeat until number is guessed:

> **Person 2 guesses a number**
>
> **Person 1 answers "correct", "too high", or "too low"**

problem size =

count :

Consider the following algorithms person 2 could use to decide the sequence of numbers to give as guesses. What is the complexity of each algorithm below?

Algorithm 1:

- Guess number = 1
- Repeat until correct:
  - If guess is incorrect, increment guess by 1

Algorithm 2:

- Guess number = N/2
- Set step = N/4
- Repeat until correct:
  - If guess is too large, next guess = guess step
  - If guess is too small, next guess = guess + step
  - step = step/2

Complexity Analysis of Number Guessing Game

Scaling the Problem Size

N	$N\log(N)$	$N2$	$2N$	$N!$
2	2.0	2	4	2
4	8.0	16	16	24
6	15.5	36	64	720
8	24.0	64	256	
10	33.2	100	1024	
15	58.6	225		
20	86.4	400		
100	664.4	10,000		
1000	9965.8	1,000,000		

Big-O Notation

constant time $\rightarrow$

linear time $\rightarrow$

quadratic time $\rightarrow$

Formal definition: