Announcements:

- Project on compbio – ok, Blackboard is the backup (and also ok)
- Midterm on Thursday, October 18 in class (will post sample tests)

From Google interview invitation:

3) The onsite interviews are somewhat similar to the phone screen in that they will cover your coding skills, analytical ability and core computer science knowledge. Topics will include algorithms, runtime/ big O/efficiency calculations problem solving, system design and of course, coding.

1

Back to Runtime Analysis

```
Public static int sumCubes( int n )
{
        int partialSum =0;

        for( int i=1; i <= n; i++ )
        {
                partialSum += i * i * i;
        }
        return partialSum;
}
```

compute sum of cubes

Declarations with initialization and return counts for constant time.
The red line is executed once for each i between 1 and n. So n times.
Total running time: linear in n.

2

General Rules

- ***for* loops**: (running time of statements inside) * (number of iterations)
- **Nested loops:** analyze inside out, runtime of inside statement * product of loop sizes

```
for  (i = 0; i < n; i++ )
        for ( j = 0; j < m; j++ )
                k++;
```

- **Consecutive statements**: add up
- ***if/else*:** runtime of the test + runtime of max of the inside statements

```
if ( conditions )
        statement1;
else
        statement2;
```

3

Remember This?

```
Public static int findMax(int [] arr)
{
        int max = arr[0];
        for( int i=1; i<arr.length; i++ )
        {
                if ( arr[i] > max ) max = arr[i];
        }
        return max;
}
```

What is the running time of the algorithm?

Now we know:
(if) x (arr.length)

4

How about This?

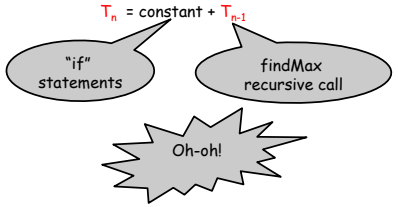
```
int max = -infinity;

Public static int findMax(int [] arr, max)
{
        if ( arr.length > 0 )
        {
                if ( arr[arr.length-1] > max )
                        max = arr[arr.length-1];
                findMax(arr[0..arr.length-2], max)
        }
        else { return max; }
}
```

What is the running time of the algorithm?

5

Running time of recursive findMax

Let T_n be the time it takes to run findMax on array of n elements.

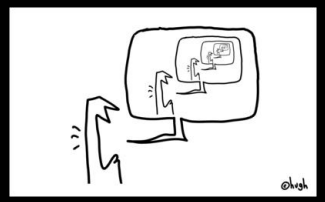


$T_n = c + T_{n-1} = c + (c + T_{n-2}) = c + (c + (c + T_{n-3})) = c + (c + (c + (c + T_{n-4}))) = \ldots$

$= c + (c + (c + (c + (\ldots(c + T_0)\ldots)))) = c*n + T_0$

Still linear!

6



CS202 Fall 2012
Lecture 12 – 10/9

Recurrences

Prof. Tanya Berger-Wolf
http://www.cs.uic.edu/bin/view/CS202/WebHome

Recursive (Inductive) Definitions

We completely understand the function f(n) = n!, right?

As a reminder, here's the definition:

$n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot (n-1) \cdot n,\ n \geq 1$

But equivalently, we could define it like this:

Recursive Case / Base Case

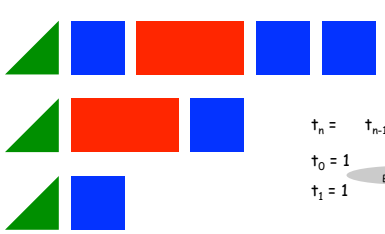
$$n! = \begin{cases} n \cdot (n-1)! & \text{if } n > 1 \\ 1, & \text{if } n = 1 \end{cases}$$

Inductive (Recursive) Definition

Recurrences

A train is defined to be an engine, followed by cars of two different kinds… long and short. Long cars are 2 units long, and short ones are 1.

Examples:



How many trains of length n are there?

$t_n = t_{n-1} + t_{n-2}$ Recursive Case

$t_0 = 1$

$t_1 = 1$ Base Case

We still don't really know.

Inductive Definitions

VERY common example:

Fibonacci Numbers

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ f(n-1) + f(n-2) & \text{if } n > 1 \end{cases}$$

Base Cases

Recursive Case

Is there a non-recursive definition for the Fibonacci Numbers?

$$f(n) = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$$

Recurrences

How many bit strings do not have 2 consecutive 0s?

Examples: 01111010, 10, 10101

$s_n = s_{n-1} + s_{n-2}$

$s_1 = 2$

$s_2 = 3$

Recurrences

A computer system considers a string of decimal digits a valid codeword if it contains an even number of 0s. Find a recurrence relation for a_n, the number of valid n-digit codewords.

Total # of codewords is sum of those starting with 0 and those starting with something else.

How many start with 0? $10^{n-1} - a_{n-1}$

How many start with something else? $9a_{n-1}$

$a_n = 10^{n-1} - a_{n-1} + 9a_{n-1} = 10^{n-1} + 8a_{n-1}$

$a_1 = 9$