

Big Oh

Intro/Motivation

What is efficiency?

(Example 1)

```
float mean(int a[], int n) {  
    sum = 0;           (line 1)  
    i = 0;            (line 2)  
    while (i < n) {   (line 3)  
        sum += a[i];  (line 4)  
        i++;          (line 5)  
    }  
    return sum / n;   (line 6)  
}
```

Def'n of big-oh

$T(n)$ is $O(f(n))$ if and only if

there exists some positive constant c such that
 $T(n) \leq c * f(n)$ for all sufficiently large values of n .

Even more formally:

$T(n)$ is $O(f(n))$ if and only if

$$\exists c, n_0 \forall n > n_0 T(n) \leq c * f(n)$$

[c and n_0 should be positive real numbers]

$T(n)$ is $O(f(n))$ if and only if

there exists some constant c such that
 $T(n) \leq c * f(n)$ for all sufficiently large values of n .

Even more formally:

$T(n)$ is $O(f(n))$ if and only if

$\exists c, n_0 \forall n > n_0 T(n) \leq c * f(n)$

More

Let's show $T(n) = 3n+4 = O(n)$

Rules of big-oh

Rules of big-oh

Why do we do this?

- Why drop coefficients and only keep the term that grows the fastest?

Examples

Categories

Graph (+ website)

Shortcuts

- You don't have to determine the exact $T(n)$ for a section of code to compute big-oh. There are shortcuts.

Loops: **(Example 2)**

```
for (int i = 0; i < n; i++) {  
    System.out.println("Hello world!")  
}
```


Shortcuts

Nested loops: **(Example 3)**

```
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < n + 25; j++) {  
        System.out.println("Hello world!")  
    }  
}
```

Shortcuts

Consecutive Statements: **(Example 4)**

```
for (int i = 0; i < n; i++)
```

```
    a[i]=0;
```

```
for (int i = 0; i < n; i++) {
```

```
    for (int j = 0; j < n + 25; j++) {
```

```
        System.out.println("Hello world!")
```

```
    }
```

```
}
```

Logarithmic time

- An algorithm takes logarithmic time --- **$O(\log n)$** --- if it repeatedly cuts the size of the problem by a constant fraction (usually $\frac{1}{2}$).
- Binary search is $O(\log n)$.

What is the tightest big-oh?

#1	<pre>sum=0; for (int i=0; i<n; i++) sum++;</pre>
#2	<pre>sum=0; for (int i=0; i<n; i++) for (int j=0; j<n; j++) sum++;</pre>
#3	<pre>sum=0; for (int i=0; i<n; i++) for (int j=0; j<n*n; j++) sum++;</pre>
#4	<pre>sum=0; for (int i=0; i<n; i++) for (int j=0; j<i; j++) sum++;</pre>

#5

```
sum=0;
for (int i=0; i<n; i++)
    for (int j=0; j<i*i; j++)
        for (int k=0; k<j+100; k++)
            sum++;
```

#6

```
x=n;
sum=0;
while(x>0){
    sum++;
    x=x/2;
}
```

#7

```
sum=0;
for (int i=1; i<n; i*=2)
    sum++;
```

#8

```
for (int i=0; i<n; i+=2){
    sum++;
}
for (int j=0; j<n; j++){
    sum++;
}
```