input array A[n]

```
for i = 1 to n
    if A[i] == 7
        for k = 1 to n
            for l = 1 to n
                print "hello"
```

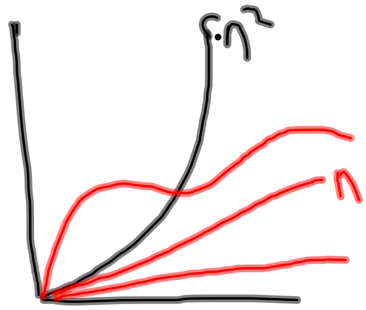
Running time?
as a function of n (input size)

Best case n

Worst case n^3

Average case?
How many 7's are in an average array?

$$O(n^2) = \left\{ \text{fns asymptotically} < n^2 \right\}$$



Claim: $2n = O(n^2)$

<u>Pf</u>: choose $c = 2$ $n_0 = 1$

Claim is $2n \le 2 \cdot n^2 \ \forall n \ge 1$

$\Leftrightarrow 1 \le n \quad \forall n \ge 1$

$\boxtimes$

<u>Claim</u> $3n+7 = O(n)$

<u>Pf</u> Choose $c=4$ $n_0=7$

$\Leftrightarrow$ $3n+7 \le 4n$ $\forall n \ge 7$

$n \ge 7$ $\forall n \ge 7$

lg

Claim $n^{\log n} = O(2^n)$

Pf $n^{\lg n} \le c\,2^n \quad \forall n \ge n_0$

$\Leftrightarrow \lg^2 n \le \lg c + n \quad \forall n \ge n_0$

$\Leftrightarrow n - \lg^2 n \ge -\lg c \quad \forall n \ge n_0$

$c = 1$

$\Leftrightarrow n - \lg^2 n \ge 0 \quad \forall n \ge n_0$

$n_0 = 4 \quad \forall n \ge 4$

at 4 $n = \lg^2 n$, n grows faster than $\lg^2 n$.

$$3n+7 = O(n)$$
$$= O(n^2)$$
$$= O(n^{11.2})$$

Running time

Given $f(n)$ what is the smallest $g(n)$

s.t. $f(n) = O(g(n))$

& $g(n)$ is "simple"

Simple

polynomial

1

$\lg n$ $\qquad$ $\lg \lg n$

$\lg^2 n$

n

$n \lg n$

n^2 $\qquad$ $n^{\lg_2 7}$

n^3 $\qquad$ $n^{\lg n}$

super-polynomial

2^n

$n!$

An algorithm runs in polynomial time if its runtime $f(n) = O(n^k)$ for some constant k.

2^n is not polynomial

False Claim: $2^n = O(n^k)$ for some k

Proof attempt: $2^n \le c n^k \quad \forall n \ge n_0$

$\iff n \le \lg c + k \lg n \quad \forall n \ge n_0$

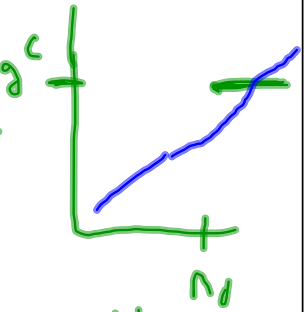
$\iff (n - k \lg n) \le \lg c \quad \forall n \ge n_0$

increasing function

limit is ∞

finite

impossible



alt. $k \lg n \le \frac{n}{2} \quad \forall n \ge 4k$

$\forall n \ge n_0$

$n \ge 4k$

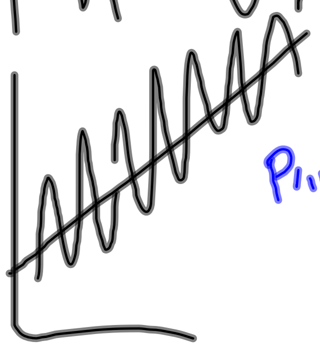
$\Rightarrow n - \frac{n}{2} \le \lg c$

$\frac{n}{2} \le \lg c$

$n^2 = \Omega(n)$

$3n+7 = \Omega(n)$

Pf: $3n+7 \geq 1 \cdot n \quad \forall n \geq 1$



$3n+7 = \Theta(n)$

$4n^2 \cdot 20 = \Theta(n^2)$

$M;M = \Theta(n^3)$

Print hello at 7 $= O(n^3)$

$= \Omega(n)$

$$\sum_{i=1}^{n} i = O(n^2)$$

for i = 1 to n
 for j = 1 to $\cancel{i}$ n
  "sing"

i = 1
 j 1
i = 2
 j 2
i = 3
 j 3

$1 + 2 + 3 + \cdots + n$

$= \frac{n(n+1)}{2} = O(n^2)$

Sep 10-5:11 PM

The sum of a decreasing geom series is O(first term)

$$n + \frac{n}{2} + \frac{n}{4} + \cdots + 1$$

$$= n\left(1 + \frac{1}{2} + \cdots + \frac{1}{n}\right)$$

$$\leq n\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right)$$

$$\leq 2n.$$

for $i = 1$ to $\lg n$

for $j = 1$ to 2^i

"dance"

$$2 + 4 + 8 + \cdots + n$$

$$n + \frac{n}{2} + \frac{n}{4} + \cdots + 2$$

$$= O(n)$$

Sep 10-5:14 PM

$\ln n = O(\lg n)$

$\lg e$

$$\lg_b a = \frac{\lg_x a}{\lg_x b}$$

$n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \frac{n}{5} + \cdots + 1 = O(n \lg n)$

$= n\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots\right)$

Sep 10-5:20 PM