Lecture 23: Time Efficiency of Algorithms

```
for (int  i = 0; i < N; i ++ )
  System.out.println (A[i]);
```

Time

Number of Items (N) →

What is this algorithm doing to array A?

```
for (int  i = 0; i < N; i ++ )
   for ( int j = 0; j < N - i - 1; j ++)
        if ( A[j] > A[j+1])
             swap ( A[j], A[j+1]);
```

Time

Number of Items (N) →

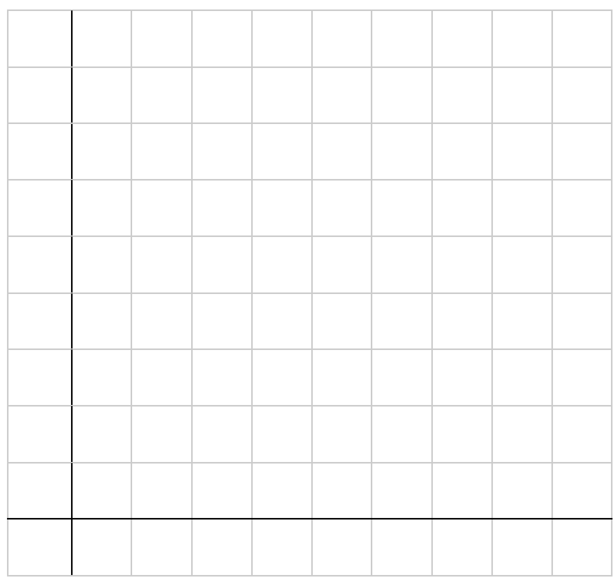
Aim: Place algorithms in categories that exhibit similar growth rates as N increases.

Graphing Functions

Real-valued functions, domain is **R** or domain is **Z**.

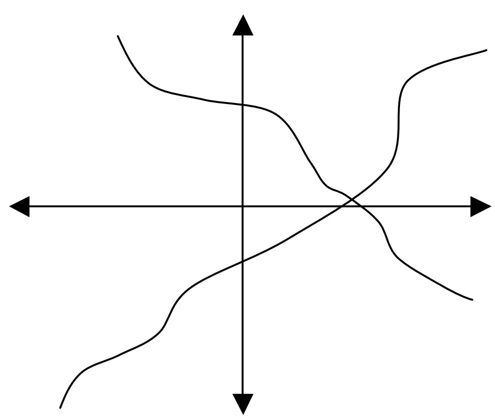
Power functions: $y = x^a$ written as $p_a(x) = x^a$

ex. $y = p_1(x) =$ $y = p_{1/2}(x) =$ $y = p_2(x) =$



Function f is **increasing** on a set S of its domain, if $\forall\, x_1$ and $x_2 \in S$ where $x_1 < x_2$, $f(x_1) < f(x_2)$.

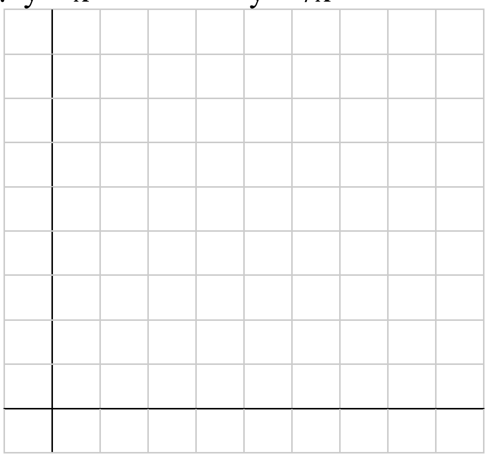
Function f is **decreasing** on a set S of its domain,



What effect does multiplying a function by a positive constant have on the shape of the graph?

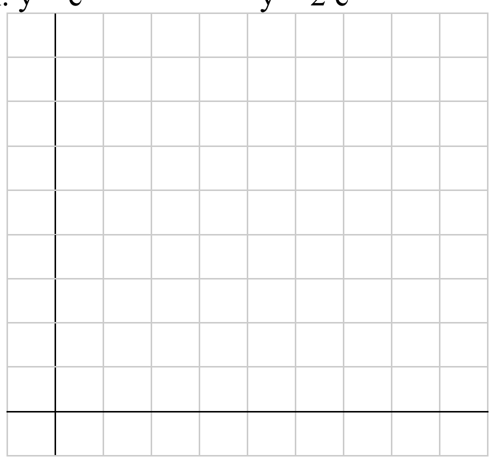
$y = f(x)$ multiply by c → $y = c\,f(x)$

ex. $y = x$ $y = 7x$

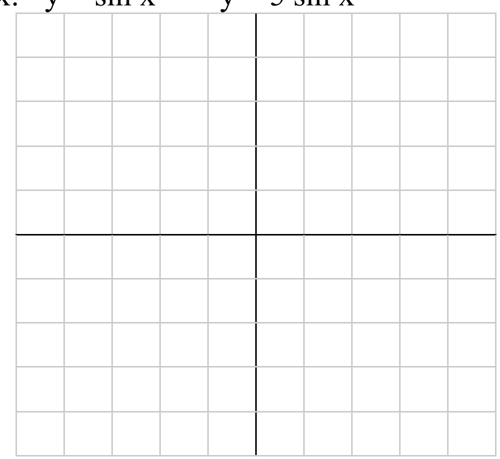


ex. $y = 3x + 1$ $y = 3\,(3x + 1)$

ex. $y = e^x$ $y = 2\,e^x$



ex: $y = \sin x$ $y = 5 \sin x$



"stretching factor" (from the x-axis)

Big-O

Def: We will say that f(x) is **O(g(x))** if there exists an x value $\mathbf{x_0}$ and some constant **M** such that

$$| f(x) | \leq \mathbf{M} \cdot | g(x) | \text{ for all } x > x_0.$$

Informally, we say that f(x) is in the O(g(x)) "family"

Note: Since in CS we are concerned with Time and time (f(x)) is always nonnegative, the inequality is

$$f(x) \leq M \cdot g(x) \quad \text{for all } x > x_0.$$

M is some "stretching factor" for g(x).

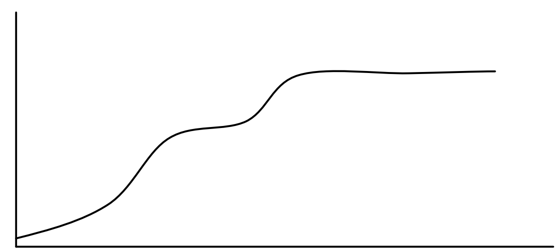
So we can "stretch" g(x) and find that the stretched version is an upper bound on ______________ for all x values that are _______________________.

Choose "simple" functions for g(x).

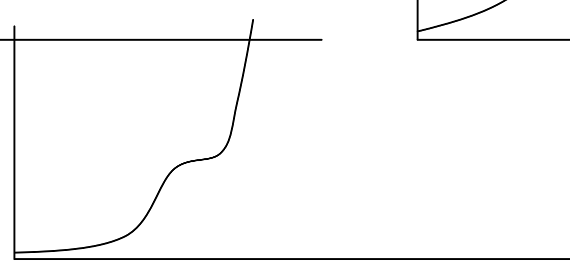
ex 1:

T

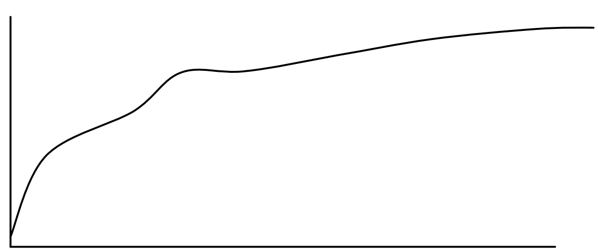
ex 2:



ex 3:



ex 4:



??? Is the choice of the g(x) for O(g(x)) unique?