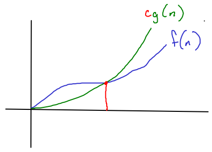
Big-O Notation

"$3n^3 + 2n^2 - 5n + 7$ is $O(n^3)$".



$\exists c > 0, k.\ \forall n \geq k.\ 3n^3 + 2n^2 - 5n + 7 \leq cn^3$

Let $c = 6$, $k = 100$. Then it suffices to show

$\forall n \geq 100.\ 3n^3 + 2n^2 - 5n + 7 \leq 6n^3$

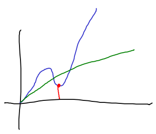
$\forall n \geq 100.\ 2n^2 - 5n + 7 \leq 3n^3$ Subtract $3n^3$ from both sides

$\forall n \geq 100.\ 2n^2 + 7 \leq 3n^3$ Because $-5n \leq 0$

ISTS

$\forall n \geq 100.\ 2n^2 \leq 2n^3$ and $\forall n \geq 100.\ 7 \leq n^3$

✓ ✓



if $a \leq b$ and $b \leq c$ then $a \leq c$

if $a \leq c$ and $b \leq d$ then $a + b \leq c + d$.

"$3n^3 + 2n^2 - 5n + 7$ is $O(n^4)$"

Generalized sorting algorithms

<

pseudocode

```
Function SelectionSort*(A, n):
    For i from 0 to n-2 inclusive:          (n-1) times
        For j from i to n-1 inclusive:      ? times
            If A[i] > A[j]:
                Swap A[i] and A[j]
            End If
        End For
    End For
End Function
```

sorted | unsorted

$n + n-1 + n-2 + \cdots$

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

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

MergeSort is $O(n \log n)$

Take stack and divide into two halves
Sort both halves
Merge two sorted halves into one stack

← Precondition: A and B are sorted

```
Function Merge(A, na, B, nb):
    C ← new array of size (na+nb)
    ia ← 0
    ib ← 0
    While ia<na and ib<nb:
        If A[ia]< B[ib]:
            C[ia+ib] ← A[ia]
            ia ← ia+1
        Else:
            C[ia+ib] ← B[ib]
            ib ← ib+1
        EndIf
    EndWhile
    While ia<na:
        C[ia+ib] ← A[ia]
        ia ← ia+1
    EndWhile
    While ib<nb:
        C[ia+ib] ← B[ib]
        ib ← ib+1
    EndWhile
    Return C
EndFunction
```

```
Function MergeSort(A, n):
    If n>1:
        X, Y ← Split(A, n)
        MergeSort(X, n/2)
        MergeSort(Y, (n+1)/2)
        Z ← Merge(X, n/2, Y, (n+1)/2)

        Copy Z into A
    Else:
```