

$$\underbrace{2n^2 + 3n + 5}_{\text{This expression}} \text{ is } O(n^2)$$

This expression grows no faster than that one except by a constant factor.

For i in 1 to n: } n times

For j in 1 to n:

a ← 0

a ← 0

Each complete execution
of whole loop: n assignments

EndFor

$$2n * n = 2n^2$$

EndFor

$$n^2 \text{ is } O(n^2)$$

$$2n^2 \text{ is } O(n^2)$$

$2n^2 + 3n + 5$ is $O(n^2)$

exists

$$\exists c \geq 1, k \geq 0, \forall n \geq k, 2n^2 + 3n + 5 \leq cn^2$$

$$c=11 \\ k=1$$

Let $c=11, k=1$. Then it suffices to show

$$\forall n \geq 1, 2n^2 + 3n + 5 \leq 11n^2.$$

$$\forall n \geq 1, 3n + 5 \leq 9n^2.$$

IF

$$a < c \\ b < d$$

then

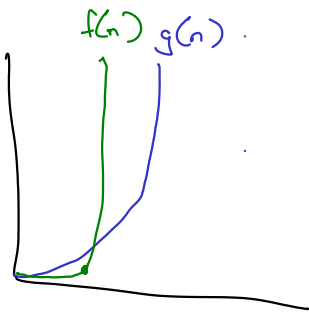
$$a + b < c + d$$

ISTS

$$\forall n \geq 1, 3n \leq 3n^2$$

and

$$\forall n \geq 1, 5 \leq 6n^2$$



$$\forall n \geq 1, 3 \leq 3n$$

$$\forall n \geq 1, 1 \leq n$$

$$\forall n \geq 1, 5 \leq 6n^2$$

QED

$$\exists c \geq 1, k \geq 0, \forall n \geq k, 2n^2 + 3n + 5 \leq cn^2$$

$$\forall n \geq 1, 1 \leq n$$

$$\forall n \geq 1, 1 \leq n$$

$$\forall n \geq 1, 3 \leq 3n$$

$$\forall n \geq 1, 5 \leq 5n \leq 5n^2$$

$$\forall n \geq 1, 3n \leq 3n^2$$

$$\hookrightarrow \forall n \geq 1, 3n + 5 \leq 3n^2 + 5n^2 \leftarrow$$

$$\forall n \geq 1, 2n^2 \leq 2n^2$$

$$\forall n \geq 1, 3n + 5 \leq 8n^2$$

$$\forall n \geq 1, 2n^2 + 3n + 5 \leq 10n^2$$

$$\exists k \geq 0, \forall n \geq k, 2n^2 + 3n + 5 \leq 10n^2$$

$$\exists c \geq 1, k \geq 0, \forall n \geq k, 2n^2 + 3n + 5 \leq cn^2$$

$$O(n^2) = O(n^2 + n)$$

$$2n^2 + 3n + 5 \text{ is } O(n^2)$$

$$2n^2 + 3n + 5 \text{ is } O(n^3)$$

Bubble Sort

2 3 2 4 7 9

Function BubSort(A, n):

For i In 0..n-1 } n times

For i In 0..n-2

If A[i] > A[i+1]:

Swap A[i] w/ A[i+1].

} O(n) } O(n²)

Merge Sort

Sorts arrays no larger than 2

Function WeakSort(A, n):

If n=0:

Finished

Else If n=1:

Finished

Else // (n=2)

If A[0] > A[1]:

Swap A[0] w/ A[1]

Function Merge(A, na, B, nb):

Precondition: A and B are sorted

Postcondition: Returns array w all elements of A & B in sorted order

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

End If

End While

While (ia < na):

C[ia+ib] ← A[ia]

ia ← ia+1

End While

While (ib < nb):

C[ia+ib] ← B[ib]

→ ib ← ib+1
End While
return C
End Function

Function MergeSort(A, n):

IF $n > 2$:

$X, nx, Y, ny \leftarrow \text{split}(A, n)$

$X \leftarrow \text{MergeSort}(X, nx)$

$Y \leftarrow \text{MergeSort}(Y, ny)$

$Z \leftarrow \text{Merge}(X, nx, Y, ny)$

Else:

WeakSort(A, n)

End IF

End Function

$O(n \log n)$