Insertion sort
Input: \( A[1..n] \) of numbers
Output: \( A[1..n] \) arranged in nondecreasing order.

\[
\text{for } i := 2 \text{ to } n \text{ do}
\]
\[
\begin{align*}
\text{save} & := A[i] \\
\text{\( j := i-1 \)} & \quad \text{while } j > 0 \text{ and } A[j] > \text{save} \\
\text{\( j := j-1 \)} & \quad \text{end while} \\
A[j+1] & := \text{save}
\end{align*}
\]
\[
\text{end for}
\]
\[
\text{\( i := i+1 \)}
\]
// A[1..n] is sorted

Argue 2 things:

1. The loop invariant holds initially (1st time loop is entered)
2. For every iteration of the loop, if the loop inv. holds at that iteration then it will also hold at the next iteration (if there is one).
Total time
\[ \sum_{i=2}^{n} (i + O(1)) \]
\[ = \sum_{i=2}^{n} i + \sum_{i=2}^{n} O(1) \]
\[ = \Theta(n^2) \]

For nested loops, work from inside out.

MergeSort:

Input: \( A[1..n] \) numbers
Output: same as before

If \( n > 1 \) then
\[ \text{mid} := \lfloor \frac{n}{2} \rfloor \]
\[ \Theta(n) \]
\[ \text{copy } A[1..\text{mid}] \text{ into new array } B[1..\text{mid}] \]
\[ \Theta(n) \]
\[ \text{copy } A[\text{mid}+1..n] \text{ into new array } C[1..(\text{mid}+1)] \]
\[ \Theta(n) \]
\[ \text{MergeSort } B[1..\text{mid}] \]
\[ \text{MergeSort } C[1..(\text{mid}+1)] \]
\[ \rightarrow \text{Merge } B \text{ and } C \text{ back into } A[1..n] \]
\[ \Theta(n) \]

Let \( T(n) \) (worst-case) running time of Merge Sort on any \( n \)-element list.
If \( n > 1 \) then

\[ T(n) = Θ(n) + T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) \]

\[ T\left(\frac{n}{2}\right) = T\left(\frac{n}{2}\right) \]

Recurrence equation

Include:

\[ T(1) = 1 \]

Prove that

\[ T(n) = Θ(n \log n) \]

by solving the recurrence equation.

\[ T(1) = 1 \]

\[ T(2) = 2c + T(1) + T(1) \]

\[ = 2c + 2 \]

\[ T(3) = 3c + T(1) + T(2) \]

\[ = 5c + 3 \]

\[ \text{Note: } n = \left\lfloor \frac{n}{2} \right\rfloor + \left\lceil \frac{n}{2} \right\rceil \]

for any \( n \geq 0 \) integer.

\[ \begin{array}{c|c|c|c|c|c|c|c}
\hline
n & cn & cn/2 & cn/2 & cn/4 & cn/4 & cn/4 & cn/4 \\
\hline
\end{array} = cn \]

\[ \text{...} \]

\[ \frac{cn}{\log n} = cn \]
\[ T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n) \]

Base case is always:

\[ T(0) = O(1) \text{ or } T(1) = O(1) \]

or both

Recursive Insertion Sort:

If \( n > 1 \) then

\[ [\text{InsertSort } A[1 \ldots (n-1)] \]

\[ n \quad \left[ \begin{array}{c} \text{Insert } A[n] \text{ into} \\ A[1 \ldots (n-1)] \end{array} \right] \]

\[ T(n) = T(n-1) + n \]

\[ T(1) = 1 \]

\[ T(n) = \sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \Theta(n^2) \]