

## Insertion Sort

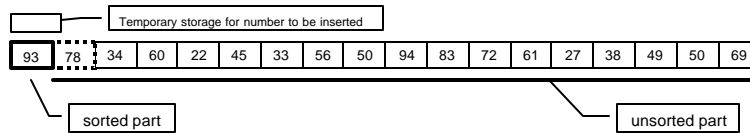
Created by Rex Woollard

Use PageUp and  
PageDown to  
move from screen  
to screen. Online  
use arrow buttons.

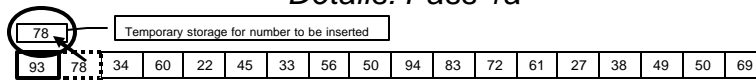
Click on  
speaker to  
play sound.



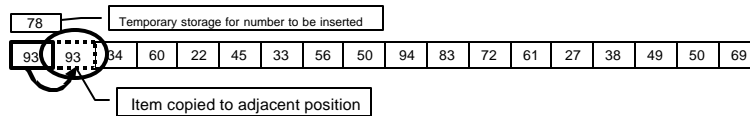
### Insertion Sort: Overview



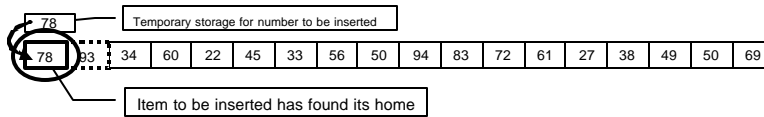
### Details: Pass 1a



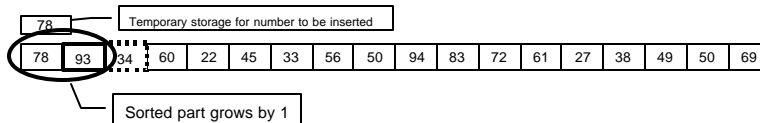
### Details: Pass 1b



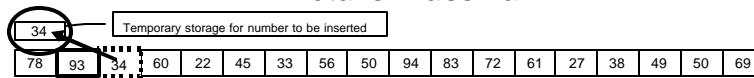
### Details: Pass 1c



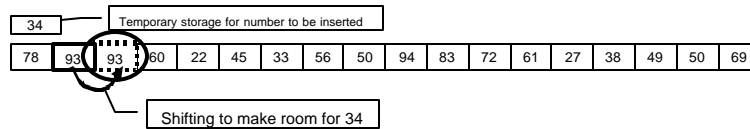
### Details: Pass 1 Finished



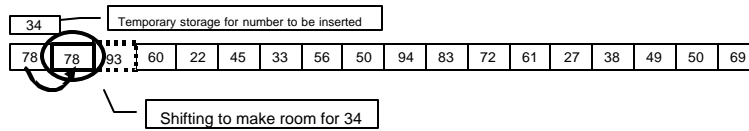
### Details: Pass 2a



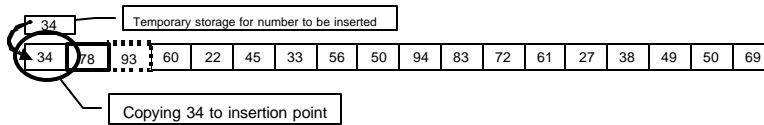
### Details: Pass 2b



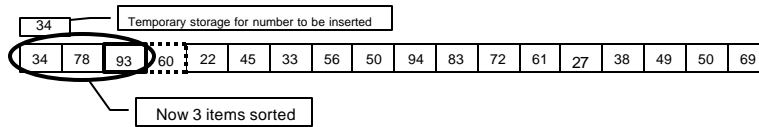
### Details: Pass 2c



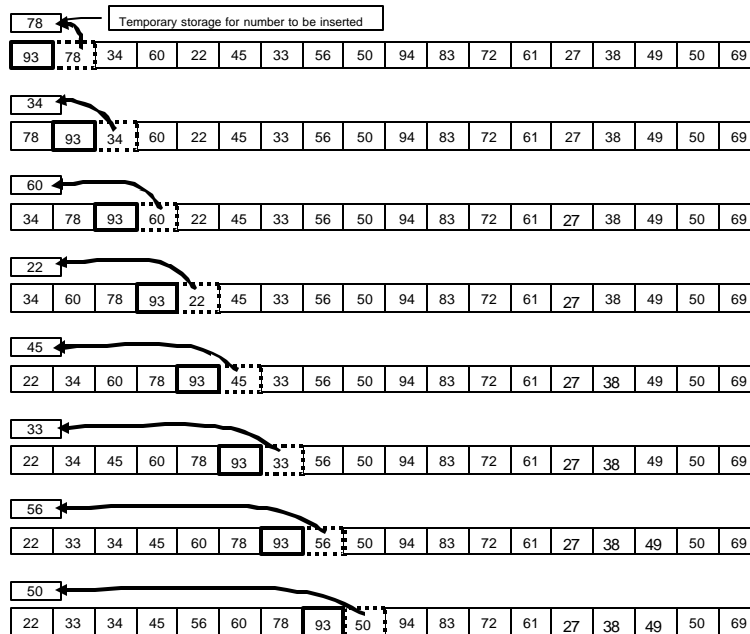
### Details: Pass 2d



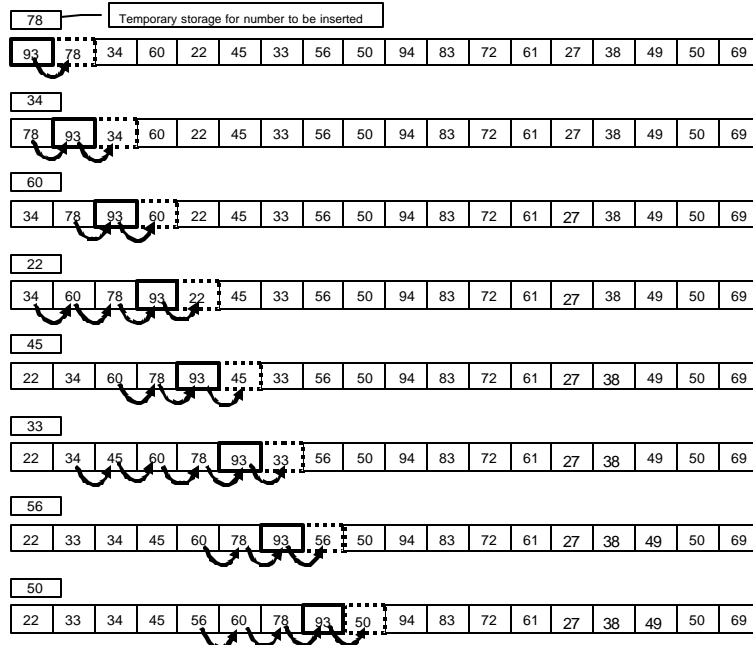
### Details: Pass 2 Finished



### Overview: Select Next Item to Insert



## Overview: Shifting to Make Room



## Overview: Place Inserted Item at Correct Location



## *Algorithm Essentials*

Conceptually, divide the list into sorted and unsorted parts.

Initial state: The first item deemed to be in correct order (since there is only one item). This suggests something about where the counter variable for your outer loop should begin.

### *Outer Loop*

Select the first item in the list of remaining unsorted elements. (This suggests where the counter variable for your inner loop should begin.) Pull this element to a temporary storage location.

#### ■ *Inner Loop*

■ Search the sorted list (one element at a time) for the correct insertion point.

■ With each comparison, another element in the sorted list is shifted one position to the right in the array to make room for the item to be inserted

Move element from temporary storage location to correct insertion point.



## *Observations about Efficiency*

Each movement of one element from the unsorted part into the sorted part is considered one pass

Given  $n$  elements, need  $n - 1$  passes

Efficiency: Two nested counting loops:  $O(n^2)$

