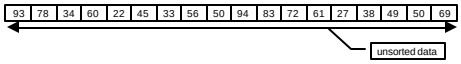
Algonquin College

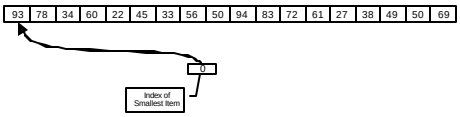
Selection Sort

Created by Rex Woollard

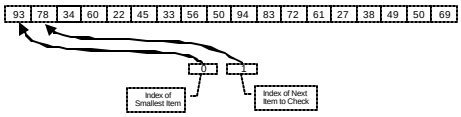
Selection Sort: Overview



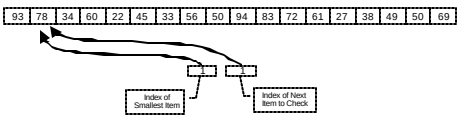
Details: Pass 1a



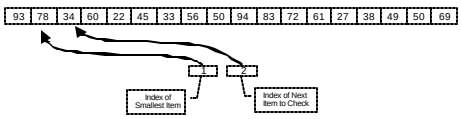
Details: Pass 1b



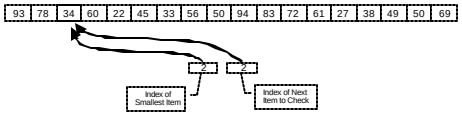
Details: Pass 1c



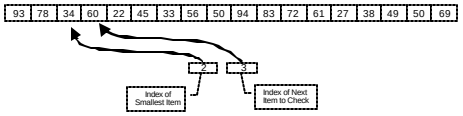
Details: Pass 1d



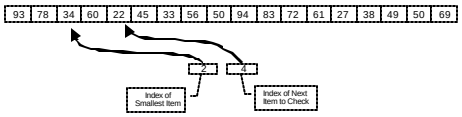
Details: Pass 1e



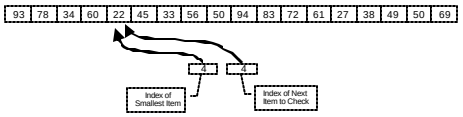
Details: Pass 1f



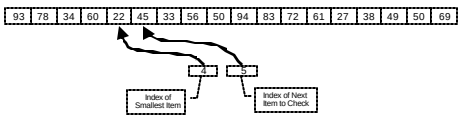
Details: Pass 1g



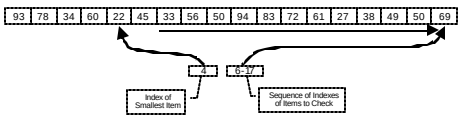
Details: Pass 1h



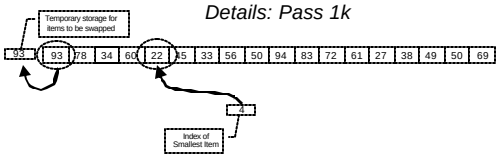
Details: Pass 1i



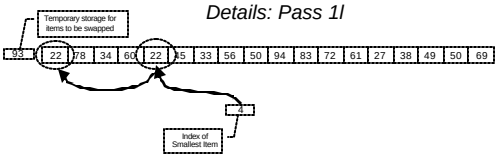
Details: Pass 1j



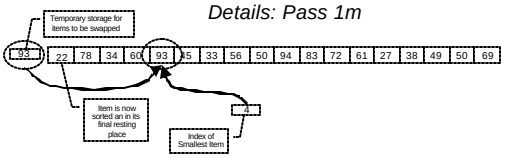
Details: Pass 1k



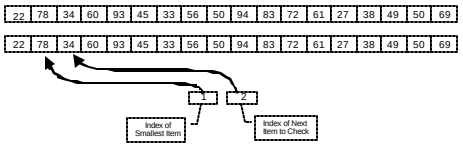
Details: Pass 1l



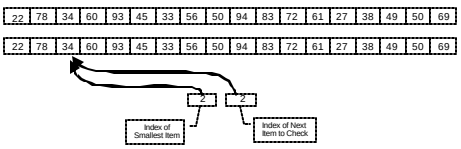
Details: Pass 1m



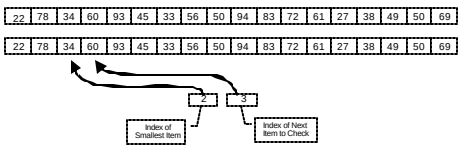
Details: Pass 2a



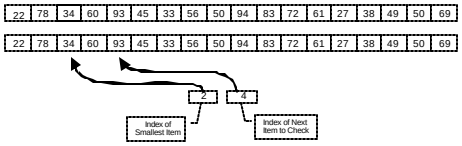
Details: Pass 2b



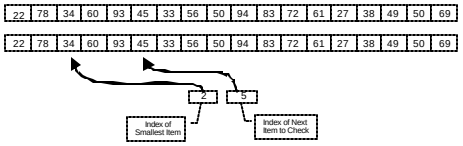
Details: Pass 2c



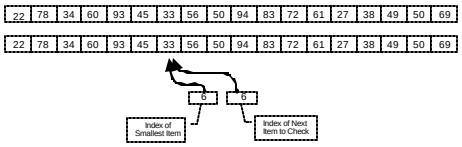
Details: Pass 2d



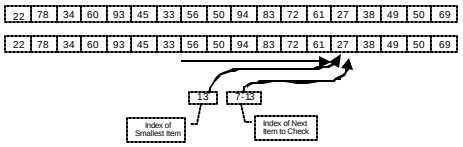
Details: Pass 2e



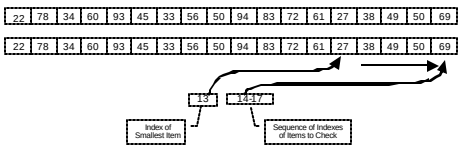
Details: Pass 2f



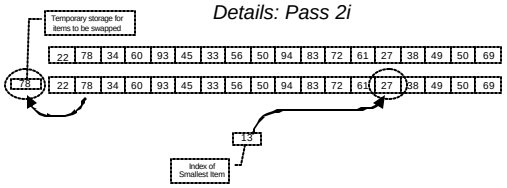
Details: Pass 2g



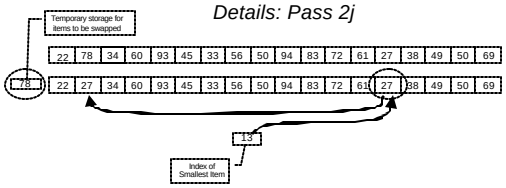
Details: Pass 2h



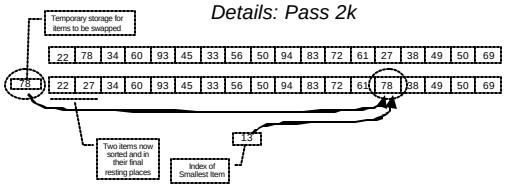
Details: Pass 2i



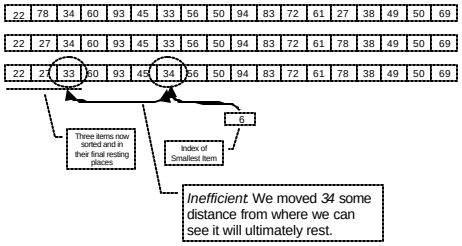
Details: Pass 2j



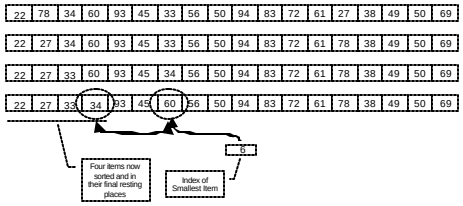
Details: Pass 2k



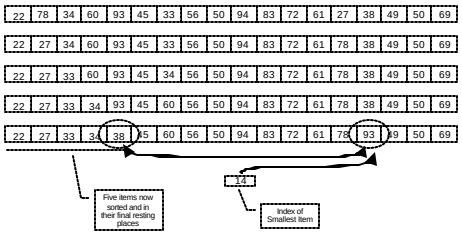
Summary: Pass 3



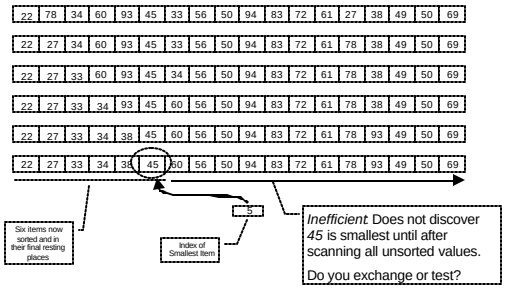
Summary: Pass 4



Summary: Pass 5



Summary: Pass 6



Algorithm Essentials

Conceptually, the list is divided into a sorted and unsorted side. Search the unsorted side for the smallest item. Move it to the sorted side.

Initial state: The list is entirely unsorted.

Outer Loop

- Manages the boundary between the sorted and unsorted.
- Arbitrarily, the first item in unsorted side is identified as the smallest item (so far). You identify the smallest using its index position within the array.
- *Inner Loop*
  - Examine all subsequent items in the unsorted side looking for an item that is smaller than the currently indexed smallest item. If one is found, set the index to smallest to the newly discovered smallest value.
  - No data movement in the *inner loop* — only setting the index to smallest.
- Following completion of the *inner loop*, swap item at index to smallest with next location in sorted side. Proceed to next iteration of *outer loop*.

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.

Each pass requires scan of all remaining unsorted elements.

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