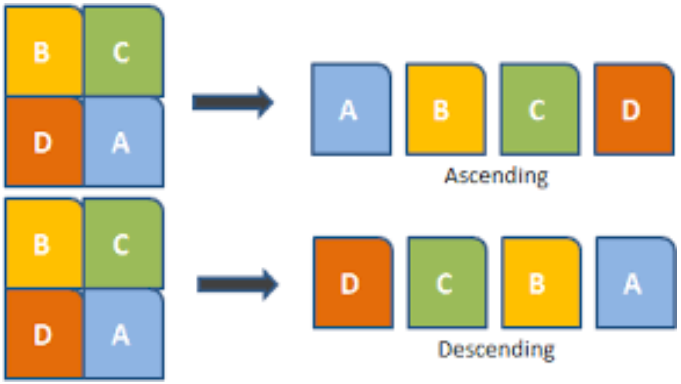
Programming with Data Structures

CS 124-04

Data Structures



Sorting

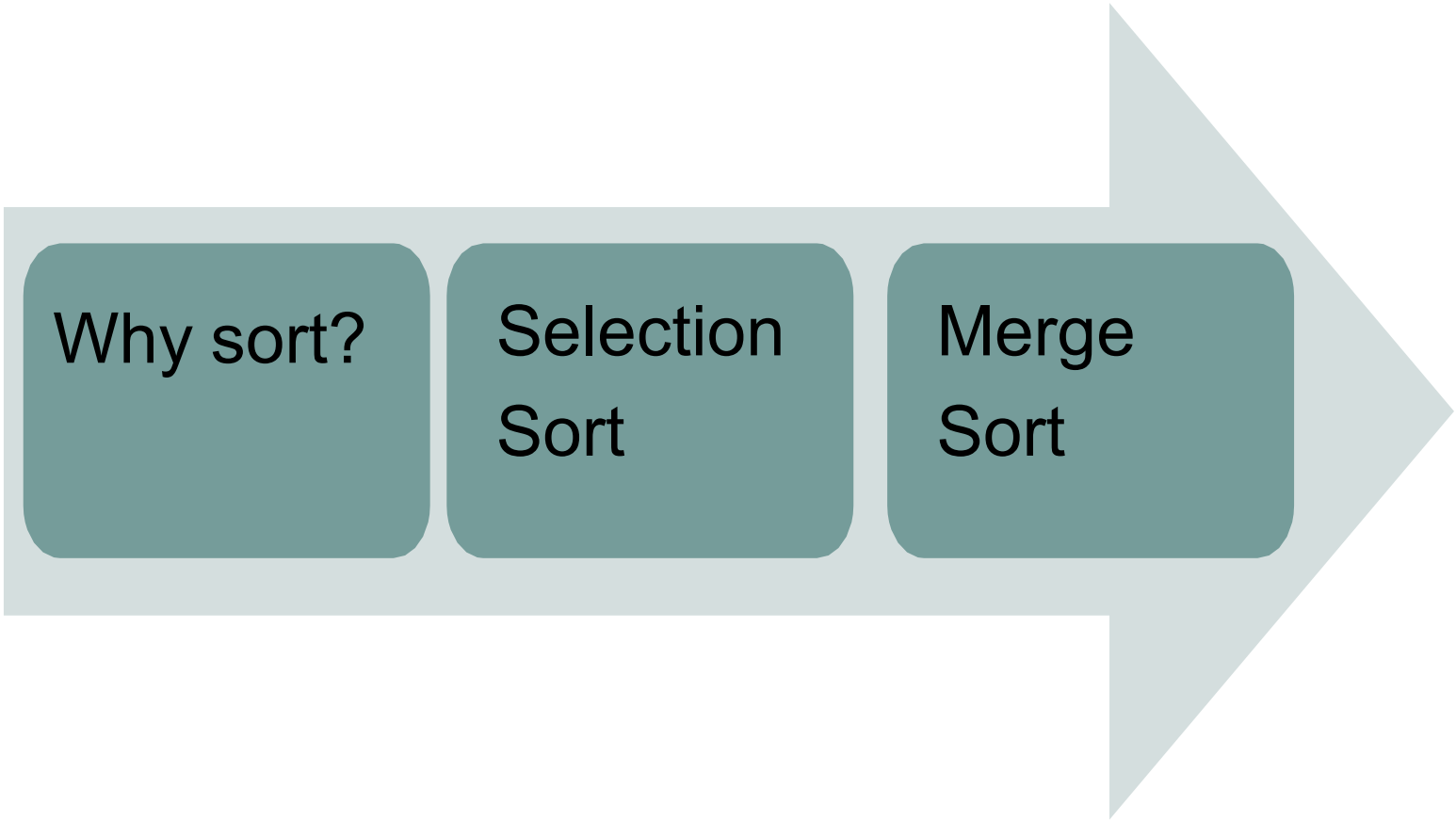
Geri Lamble

CS 124-03

Reading this Week

Text: Chapter 12.1-12.5

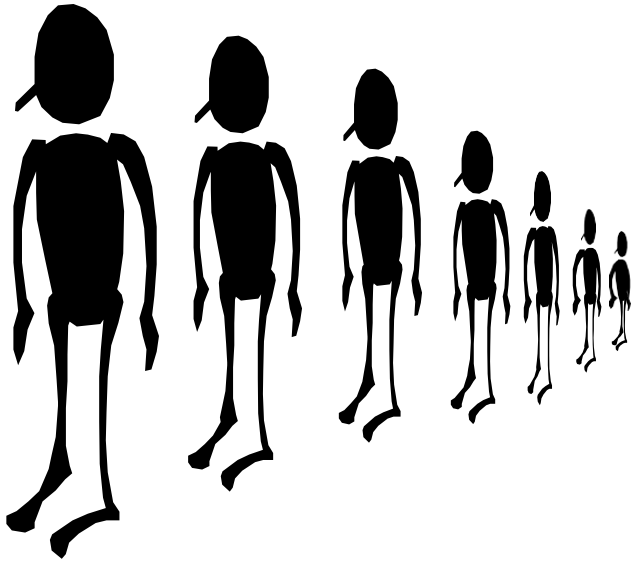
Outline



Why Sort?

Sorting Algorithms

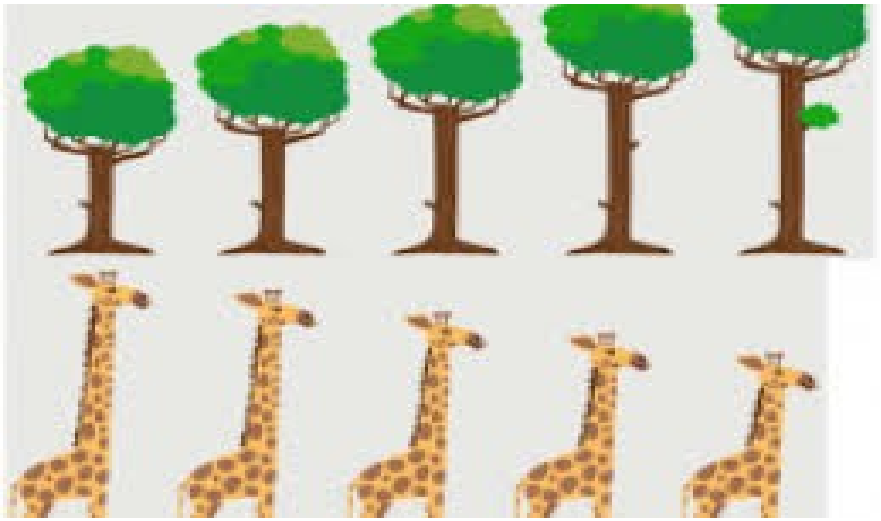
Sort



They start with a potentially unsorted array and return with an array whose contents are sorted.

Sorting Order

Ascending – 1st element in array is the smallest.



Descending – 1st element in array is the largest.

Ideal Sorting Algorithm

Introduce some terminology and concepts

- Stable: equal keys are not re-ordered. Preserves left to right ordering of redundant numbers from an unsorted array.

  Example:

  5 1 2 3 4 5

  1 2 3 4 5 5

- Extra Space: Operates in place.
- Key Comparisons: Takes theoretical minimum of (O(nlogn)).
- Swaps: Worst case O(n) number of swaps.
- Adaptive: Improves execution time if sorted data.

Selection Sort

Overview

- Finds smallest* value in the list and swaps with first value.
- Then starts on second value repeats above process.
- Repeats this process until all necessary replacements have been made.

*smallest sort increasing order

or

*largest sort decreasing order

Selection Sort Example

- Given a list of these values:

11 9 4 3 15 13 14
3 9 4 11 15 13 14
3 4 9 11 15 13 14
3 4 9 11 13 15 14
3 4 9 11 13 14 15

***Sort increasing order.**

Selection Sort Algorithm 1

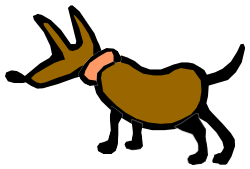
1st. Find the index of the largest element and exchange the position with the element at the last index.

0	1	2	3	4	5	6	7	8
60	30	10	20	40	90	70	80	50

0	1	2	3	4	5	6	7	8
60	30	10	20	40	50	70	80	90

2nd, Decrement the last index.

When the search index is exhausted, the sorting stops.



0	1	2	3	4	5	6	7	8
10	20	30	40	50	60	70	80	90

Data Independence

The best case of sorting:
1,2,3,4,5,6,7
The worst case of sorting:
7,6,5,4,3,2,1

But with the selection sort method, the time complexity for each case is the same, because whatever the array is looks like, the function will evaluate all values.
So, the number of comparisons and assignments is independent of the data.

Time Complexity

- O(n^2) for worst, best, and average.
- Execution time: Because of no ending condition, sort always goes through every remaining element.
- Execution space: Because one swap is made with each passing of list.
- Elements needing replacement in list shrinks by one after each passing.
- So, with a list of n length the complexity looks like: n + (n-1) + (n-2) + ... + 1.

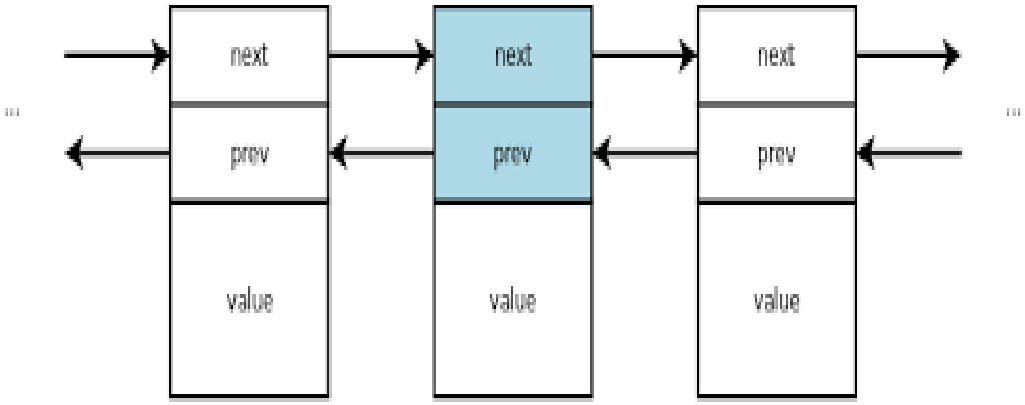
Linked List Variant

- When minimum value is found, it is placed at front of list.
- Then the replaced element is put at the end of the list.
- Process continues until remainder of list is stepped through.

Linked List Example

- Given a list of values:

7 4 1 3 6
1 4 3 6 7
1 3 6 7 4
1 3 4 7 6
1 3 4 6 7



Sort Efficiency

- At most, sort steps through the entire list once for each element in list.
- Extremely inefficient sort algorithm for lengthy lists.

Strengths and Advantages

- Simple algorithm.
- No additional memory nor data structures are needed.
- Elements are swapped and not copied into a holding cell.

Merge Sort

What is Merge Sort?

"Merge sort" is an algorithm that works by breaking an unsorted list of length *n* into *n* sublists, and then merges them with their adjacent lists until there is only one, sorted list remaining.

It has a complexity of O(n logn).

Merge Sort

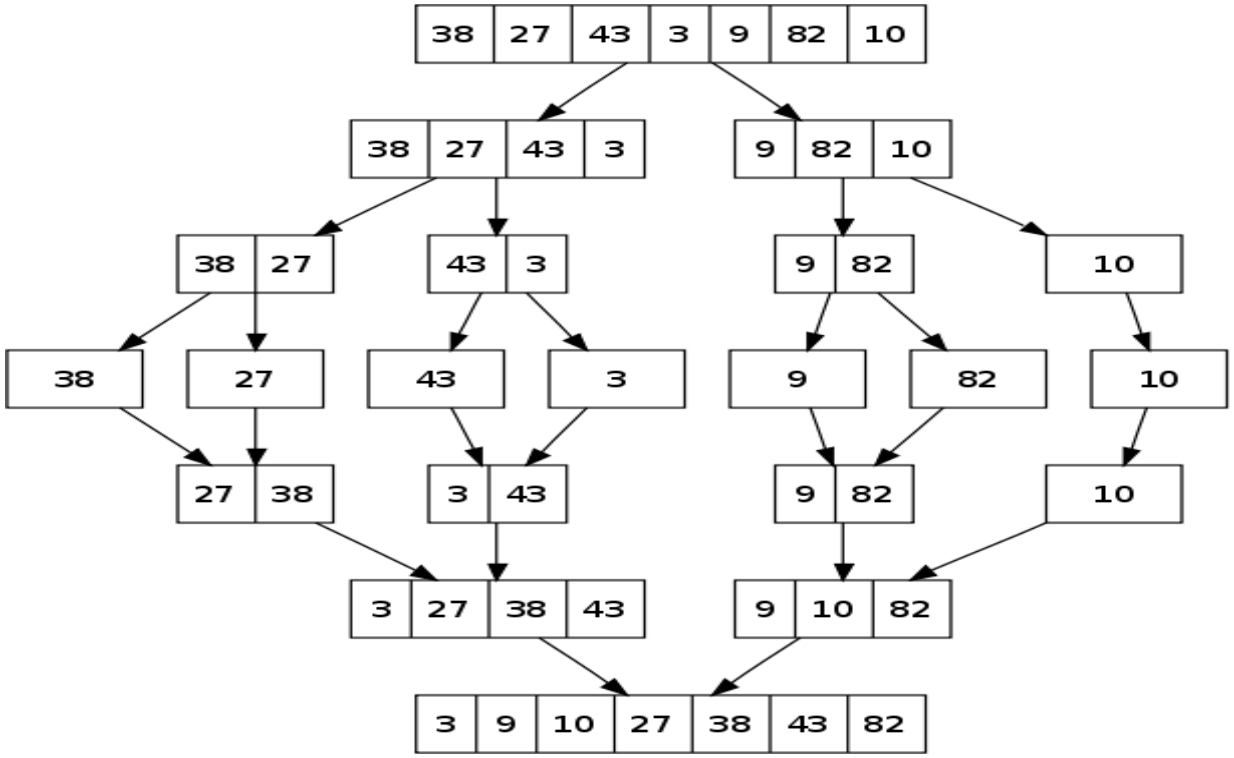
Example 1:

List of length 4: [5,3,9,6]
List is broken up into 4 lists: [5], [3], [9], [6]
First list is merged with second list: [3,5]
Third list is merged with fourth list: [6,9]
Lists are merged together: [3,5,6,9]

Merge Sort

Example 2:

As the length of the list increases, the process becomes far more complex



Merge Sort

Running Time

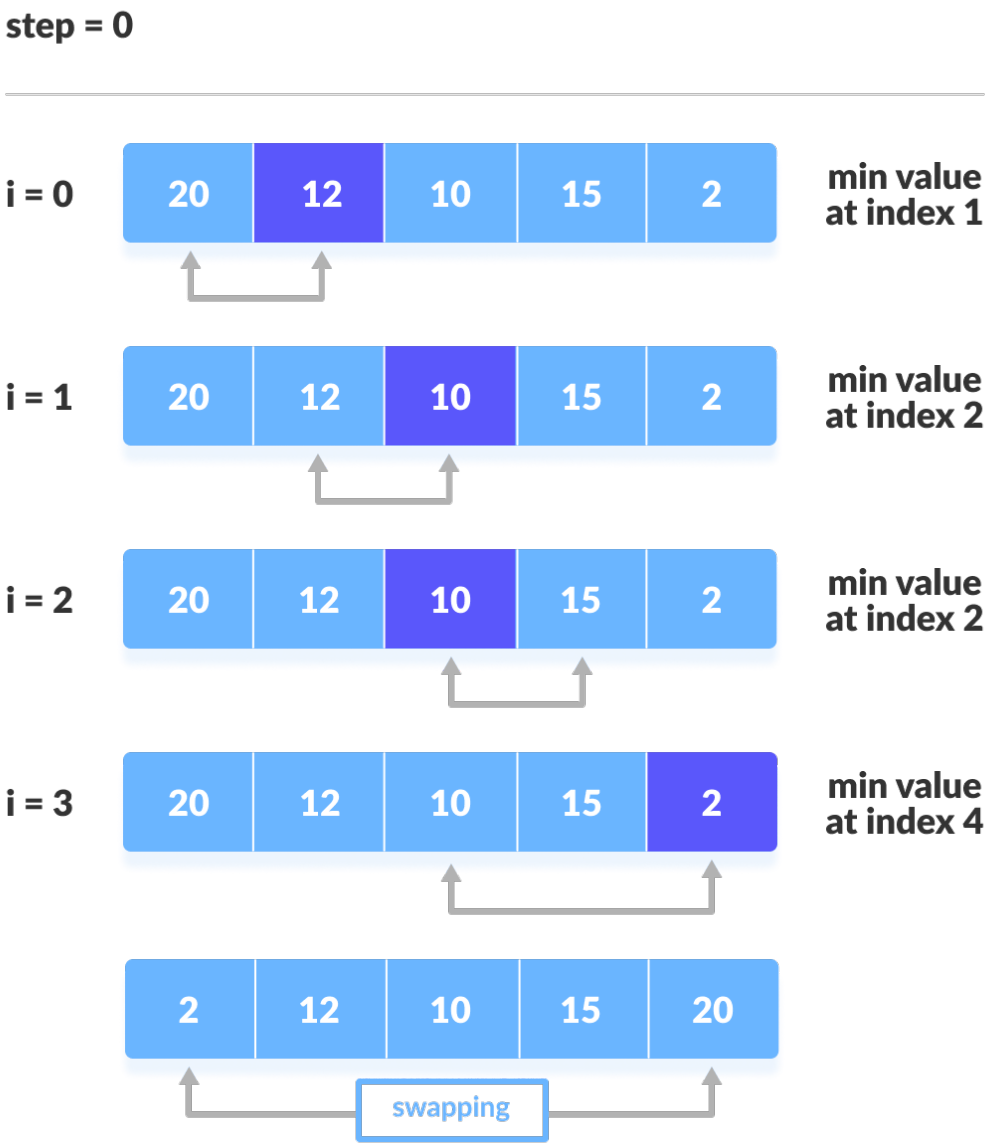
- Best Case: O (n log n)
- Worst Case: O (n log n)
- Average Case: O (n log n)

Merge Sort Efficiency

- Most Merge Sort implementations do not run in-place.
- In order to run in-place, or without extra memory, use recursive calls to implement Merge Sort.

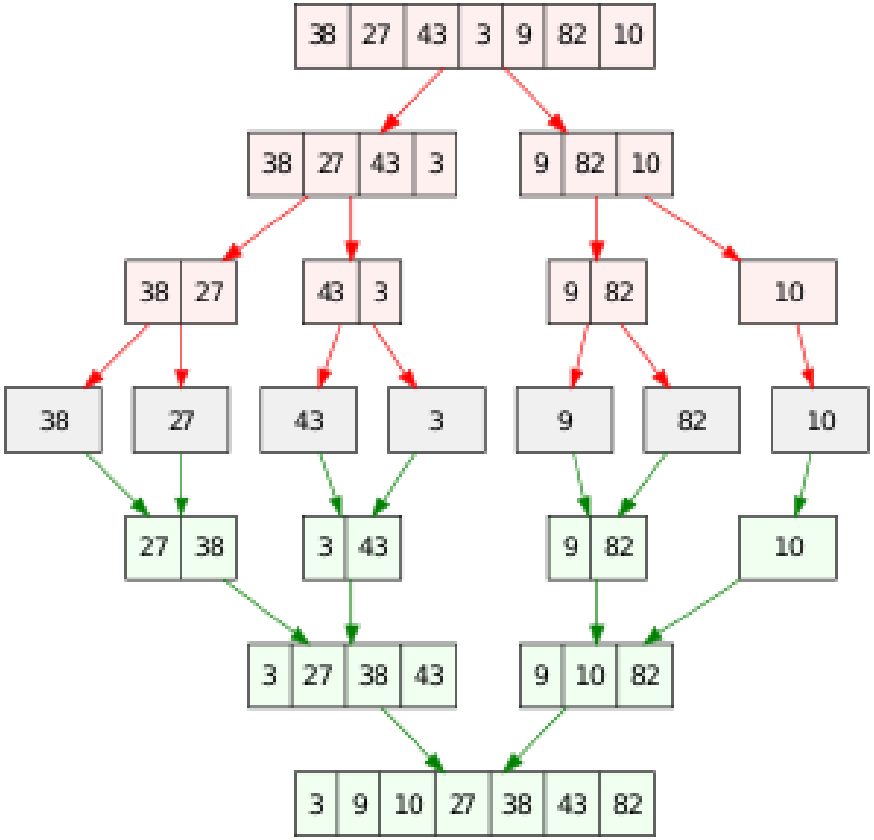
Reference

Selection Sort



References

[Merge Sort]



Up Next

DATA STRUCTURE