Algorithms and Sorting

Course: CPSC 1150

Instructor: Dr. Bita Shadgar

Lecture 22

Learning Outcomes

- Define and apply algorithm for sorting a list

Sorting an array

Example

Sort an array of numbers (called myList) in ascending order

- Problems in this form show up everywhere in computer science
- Often its not numbers, but other comparable objects (anything that can be ordered)
- Sometimes its descending order instead
- No matter the details, the same algorithms can be used with many different types of data, in a variety of applications
- Question: If you had to sort something manually (assignments by last name, books by title, pens by color) how would you do it?

Some sorting algorithms

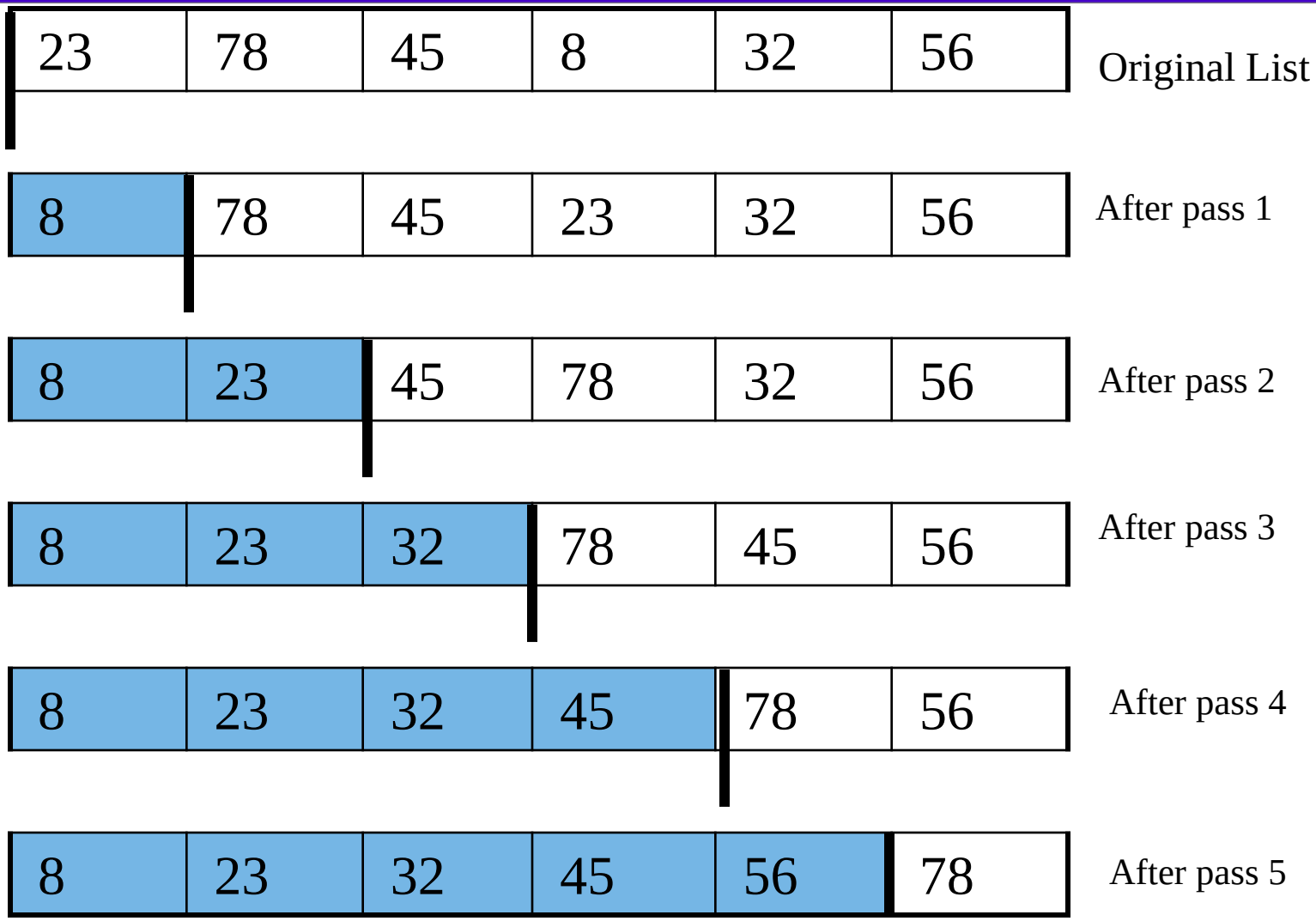
There are many different sorting algorithms out there

- Bubble sort
  - Swaps neighboring elements until everything is in order
- Selection sort
  - Selects min element and swaps with current element
- Insertion sort
  - Takes the next element and inserts into the sorted portion
- Merge sort
  - Sort chunks of the list and merge the chunks
- Quick sort
  - Organizes elements based on pivot values And more. . .
- To see some very neat visualizations of these algorithms, and learn more, check out:
  - http://www.sorting-algorithms.com/
  - https://visualgo.net/en/sorting

Selection Sort

- For an array with n elements, the idea is as follows:
  1. Find the minimum element in the list and swap it with the first
  2. Find the minimum element among the remaining n − 1 elements, and swap it with the second
  3. Find the minimum element among the remaining n − 2 elements, and swap it with the third
  4. etc.
- One of the simplest sorting algorithms
- Performs very many comparisons but very few swaps

Selection Sort



Pseudo-code : Selection sort

Selection Sort

1. START
2. Set i = 0
3. WHILE i < n − 1, REPEAT:
    3.1 Set j = the index of the minimum element in $\{L_i, \ldots, L_{n-1}\}$
    3.2 IF j ≠ i:
        3.2.1 Swap positions j and i in L
    3.3 Increment i by 1
4. END

Swap

1. START
2. Set temp = K_i
3. Set K_i = K_j
4. Set K_j = temp
5. END

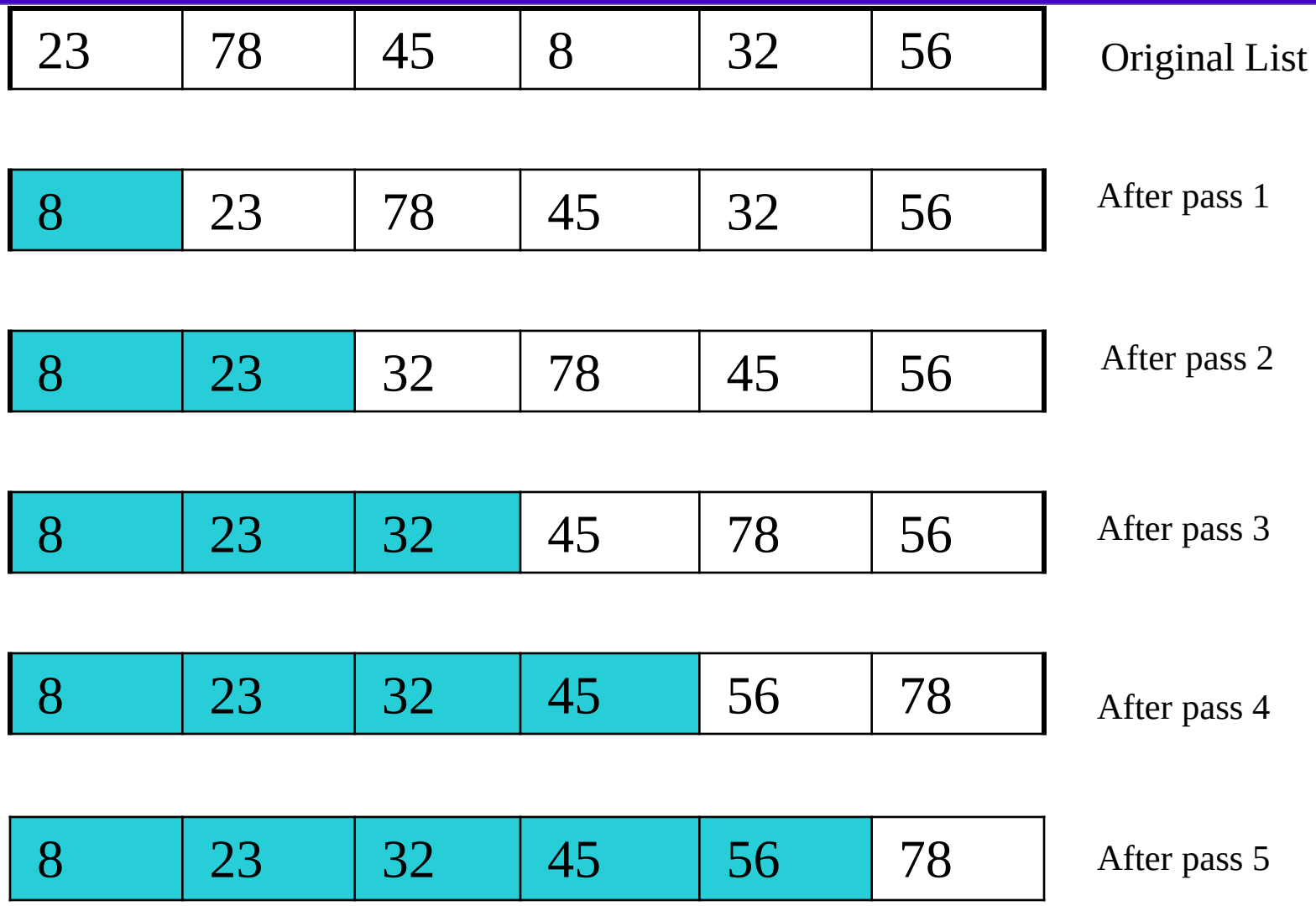
Finding index of the minimum

1. START
2. Set min = i
3. WHILE i < n, REPEAT:
    4.1 IF K_i < K_{min}:
        4.1.1 Set min = i
    4.2 Increment i by 1
5. Output min
6. END

Bubble Sort

- Bubble sort uses the same selection sort approach:
  - Find the min/max item
  - Put it into its proper place
- But a different scheme is applied for finding the min/max item:
  - Starting with the last (first) item, compare successive pairs of items, swapping whenever the bottom item is smaller than the one above it

Bubble Sort


23 78 45 8 32 56
Original List
8 23 78 45 32 56
After pass 1
8 23 32 78 45 56
After pass 2
8 23 32 45 78 56
After pass 3
8 23 32 45 56 78
After pass 4
8 23 32 45 56 78
After pass 5

Pseudo-code : Bubble sort

Bubble Sort

```
1. START
2. Set i = 1
3. WHILE i < n , REPEAT:
    3.1. for j from n-1 to i
        3.1.1. if a[j - 1] > a[j]
            3.1.1.1. swap(a[j], a[j -1] )
    3.2. Increment i by 1
4. END
```

Swap

1. START
2. Set temp = K_i
3. Set K_i = K_j
4. Set K_j = temp
5. END

Bubble Sort – How to improve it?