



Sorting

Mais S. Al-Saoud
Computer Science Department
Karbala University

Jan 2016



What is sorting?

- Sorting means “Putting things in Order” or “Arranging”
- Arranging can be done in two ways
 - Ascending Order (Smallest to Largest)
 - Descending Order (Largest to Smallest)
- In default Sorting means arranging in ascending order
- There are many algorithms for sorting, some simple but perhaps inefficient others complicated but efficient.



Sorting algorithms

- @ Bubble sorting
- @ Selection sorting
- @ Insertion sorting
- @ Shell sorting
- @ Merge sorting
- @ Quick sorting



Sorting Applications

- To arrange a set of items in sequence
- To prepare a list of student ID, names, and scores in a table (sorted by ID or name) for easy checking.
- Efficient sorting is important for optimizing the use of other algorithms such as searching

Selection Sorting

Working : "SELECT" an Element and Put it in PROPER PLACE

Description :

1. From position 0, find the smallest and then exchange it with the element in position 0.
2. From position 1, find the smallest and exchange with position 1.
3. Now from 2 do the same until you have reached the end of the list



Selection Algorithm

$i=0$

Find smallest element from $\text{data}[i]$ to $\text{data}[n-1]$

Interchange the smallest with $\text{data}[i]$

Increment i

Repeat for $i=n-1$

stop

Selection Ex1

67	33	21	84	49	50	75	INITIAL State
21	33	67	84	49	50	75	pass1
21	33	67	84	49	50	75	Pass 2
21	33	49	84	67	50	75	Pass 3
21	33	49	50	67	84	75	Pass 4
21	33	49	50	67	84	75	Pass 5
21	33	49	50	67	75	84	Pass 6

Selection Ex2

20	8	5	10	7
----	---	---	----	---



5	8	20	10	7
---	---	----	----	---



5	7	20	10	8
---	---	----	----	---



5	7	8	10	20
---	---	---	----	----

5	7	8	10	20
---	---	---	----	----



Bubble Sort

- Bubble sort examines the array from start to finish, comparing elements as it goes.
- Any time it finds a larger element before a smaller element, it swaps the two.
- In this way, the larger elements are passed towards the end.
- The largest element of the array therefore "bubbles" to the end of the array.
- Then it repeats the process for the unsorted portion of the array until the whole array is sorted.

(compare j with j+1 and swap)

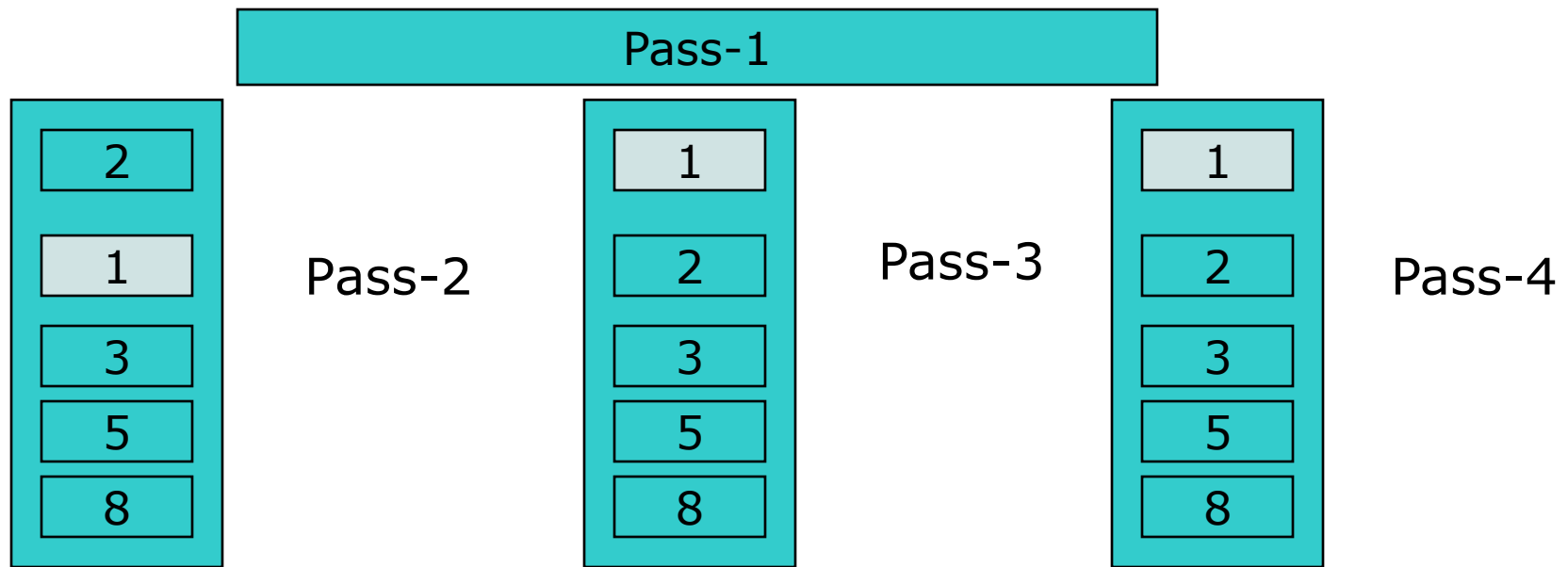
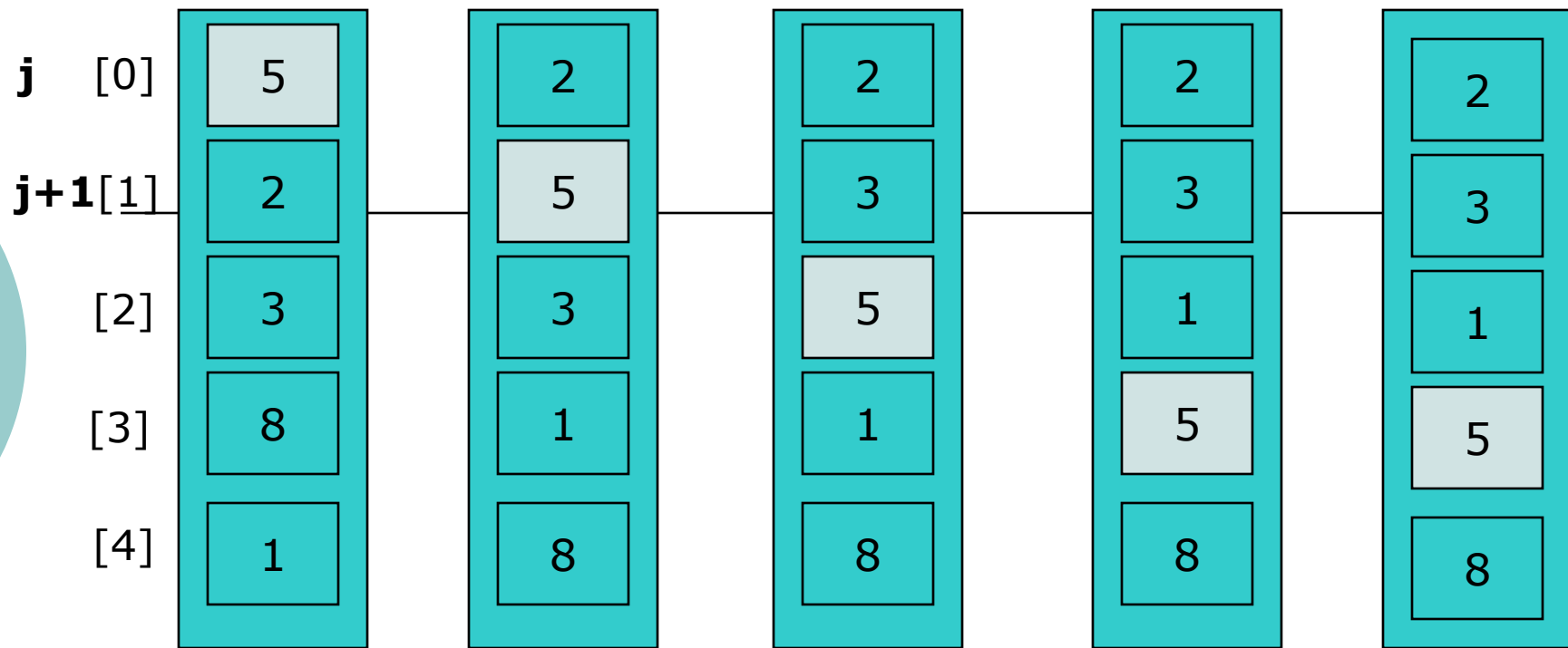


FIGURE 10.1

One Pass of Bubble Sort

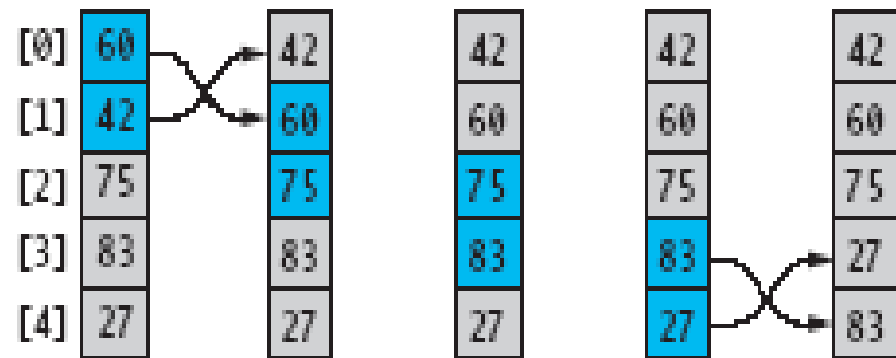
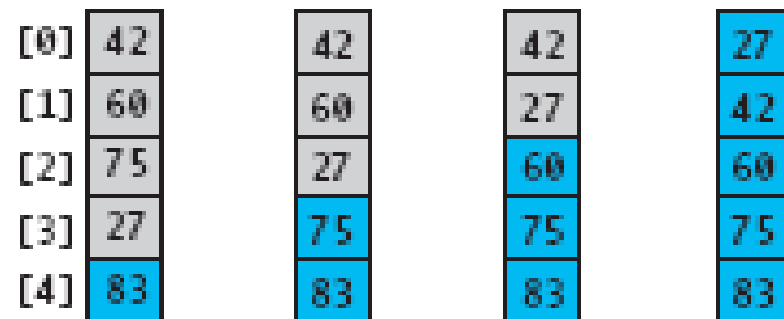


FIGURE 10.2

Array After Completion
of Each Pass



Bubble sorting

- Working:

It works by comparing neighbours in the array and exchanging them if necessary to put the smaller of the pair first.

On each pass through the array an element 'bubbles' up into place.

- General Algorithm:

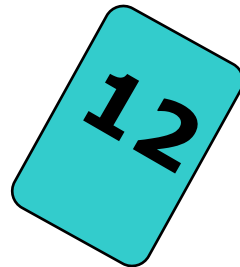
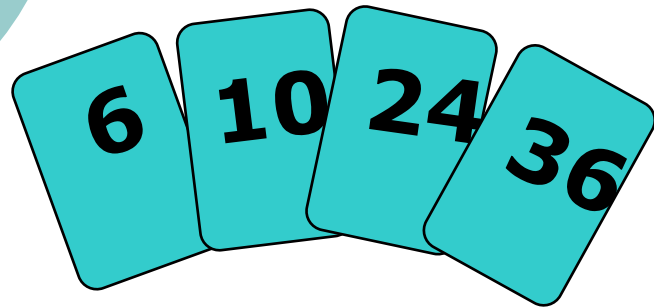
```
for (i=0;i<max;i++)  
    for (j=0;j<max-1-i;j++)  
        if (a[j]>a[j+1] )  
            interchange a[j] and a[j+1]  
                save = a[j]  
                a[j] = a[j+1]  
                a[j+1] = save
```



Insertion Sort

- **Idea:** like sorting a hand of playing cards
 - Remove one card at a time from the table, and insert it into the correct position in the left hand
 - compare it with each of the cards already in the hand, from right to left
 - The cards held in the left hand are sorted

Insertion Sort



To insert 12, we need to make room for it by moving first 36 and then 24.

Insertion Ex1

temp

8

20 8 5 10 7

i = 1, first iteration

8

20 20 5 10 7

8 20 5 10 7



Insertion Ex1

temp

5

8 20 5 10 7

i = 2, second iteration



5

8 20 20 10 7



5

8 8 20 10 7



5 8 20 10 7

Insertion Ex1

temp

10

5 8 20 10 7

i = 3, third iteration



10

5 8 20 20 7



5 8 10 20 7

Insertion Ex1

temp

7

5 8 10 20 7

i = 4, forth iteration

7

5 8 10 20 20

7

5 8 10 10 20

7

5 8 8 10 20

5 7 8 10 20



Insertion Ex2

67	33	21	84	49	50	75
67						
33	67					
21	33	67				
21	33	67	84			
21	33	49	67	84		
21	33	49	50	67	84	
21	33	49	50	67	75	84

Insertion sorting

Working:

The Elements are Picked one by one and put into the Proper Place

Insertion_Sort (int array[],int size)

```
{
    int i,j,index;
    for (i=1;i<size;i++)
    { index = array[i];
      j=i;
      while ((j>0)&&(array[j-1]>index))
      { array[j] = array [j-1];
        j=j-1;}
      array[j] = index;
    }
```