PROGRAMMING FOR PROBLEM SOLVING -- PPS

SUBJECT CODE-BTPS-101-18

By
Dr.A.Deepa,AP,Chandigarh Engineering College

PPS SYLLABUS

Unit 4

Basic Algorithms (6 lectures)

Searching, Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, notion of order of complexity through example programs (no formal definition required)

Sorting

- The process of arranging the elements in a list in either ascending or descending order.
- Types of Sorting:

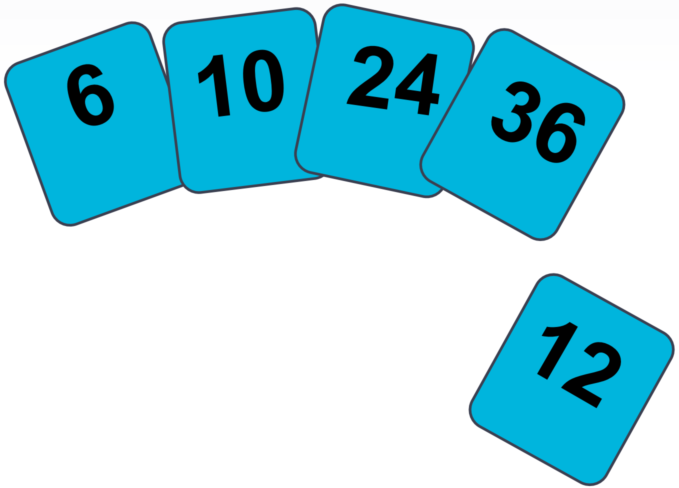
- *Bubble Sort*
- *Insertion Sort*
- *Selection Sort*
- *Merge Sort*
- *Quick sort*
- *Heap Sort*
- *Radix Sort*

By Dr.A.Deepa,AP,Chandigarh Engineering College

Insertion Sort

- Idea: like sorting a hand of playing cards
- Start with an empty left hand and the cards facing down on the table.
- 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
- these cards were originally the top cards of the pile on the table

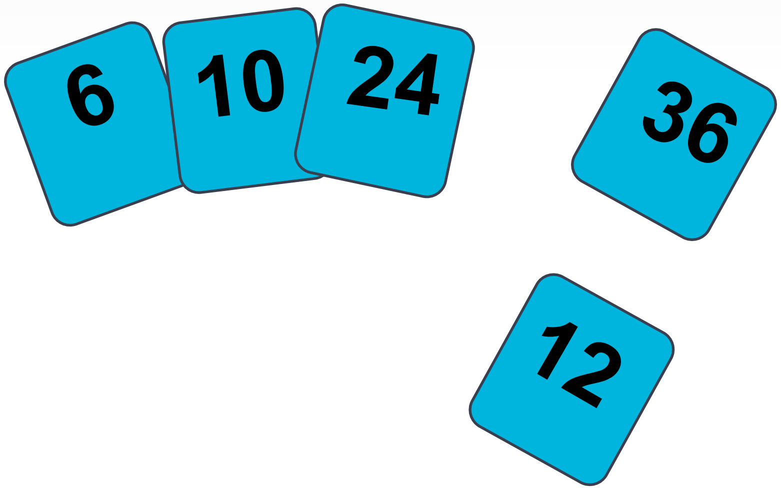
Insertion Sort



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

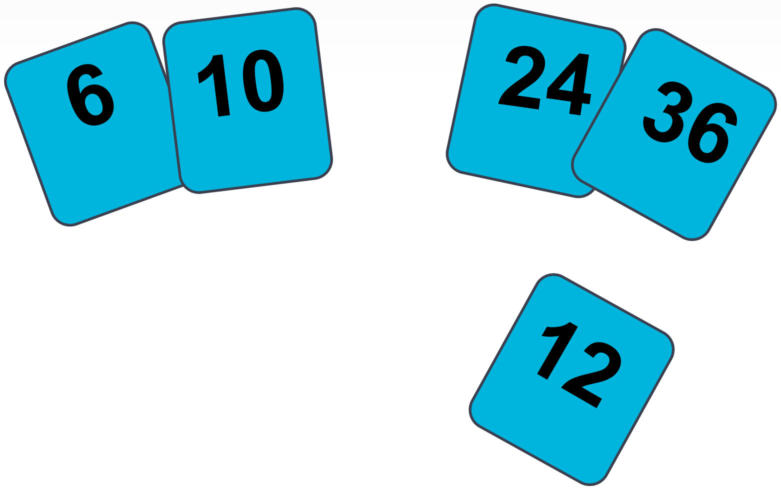
By Dr.A.Deepa,AP,Chandigarh Engineering College

Insertion Sort



By Dr.A.Deepa,AP,Chandigarh Engineering College

Insertion Sort



By Dr.A.Deepa,AP,Chandigarh Engineering College

Insertion Sort

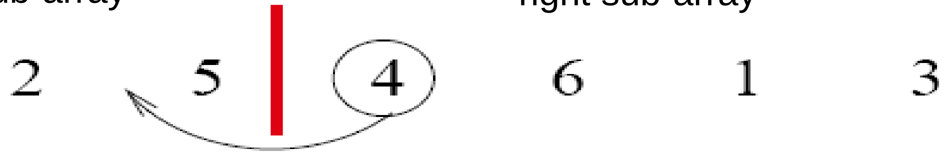
input array

5 2 4 6 1 3

at each iteration, the array is divided in two sub-arrays:

left sub-array

right sub-array

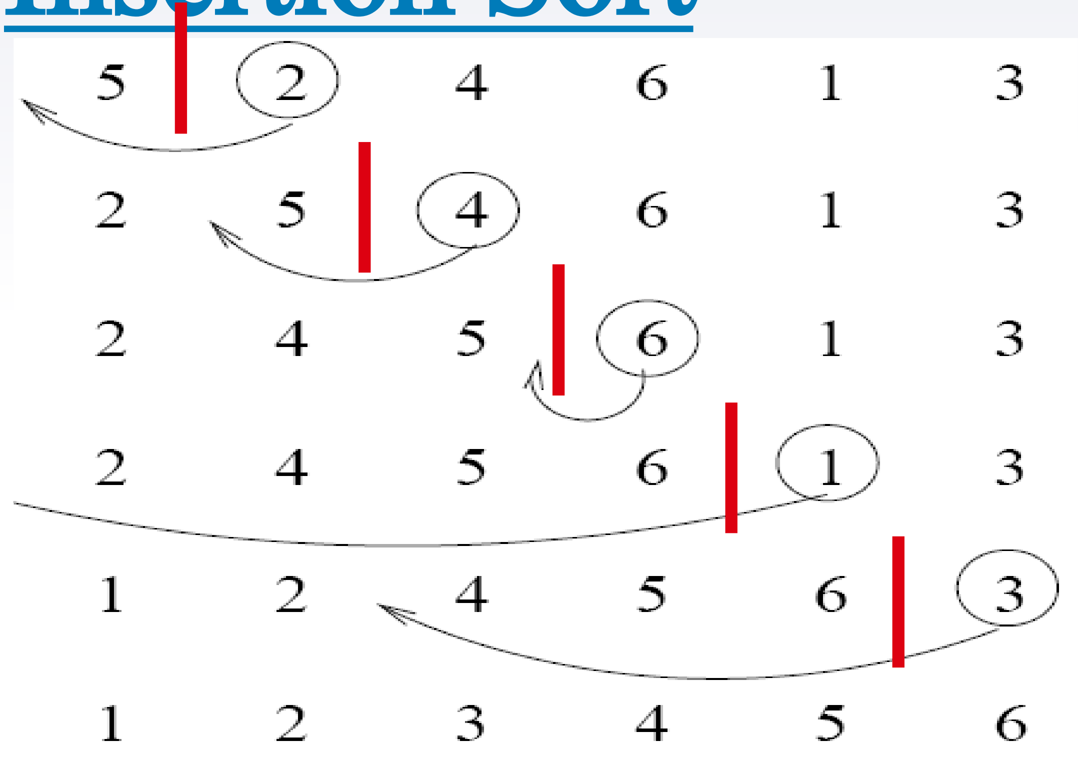


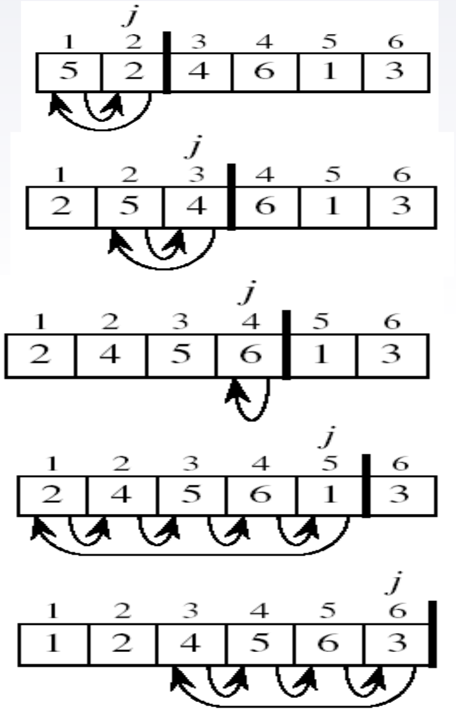
sorted

unsorted

By Dr.A.Deepa,AP,Chandigarh Engineering College

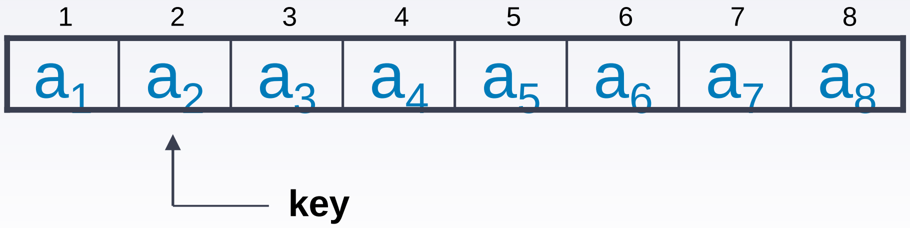
Insertion Sort





By Dr.A.Deepa,AP,Chandigarh Engineering College

INSERTION-SORT

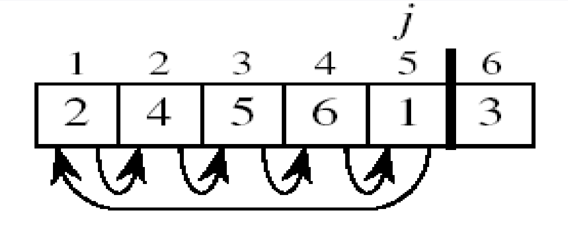


Alg.: INSERTION-SORT*(A)*

```
for j ← 2 to n
    do key ← A[ j ]
        Insert A[ j ] into the sorted sequence A[1 . . j -1]
       i ← j - 1
       while i >= 0 and A[i] > key
            do A[i + 1] ← A[i]
                i ← i – 1
       A[i + 1] ← key
```

- Insertion sort – sorts the elements in place

Loop Invariant for Insertion Sort



Alg.: INSERTION-SORT(A)

for j ← 2 to n

do key ← A[j]

Insert A[j] into the sorted sequence A[1 . . j -1]

i ← j - 1

while i > 0 and A[i] > key

do A[i + 1] ← A[i]

i ← i – 1

A[i + 1] ← key

Invariant: at the start of the for loop the elements in A[1 . . j-1] are in sorted order

By Dr.A.Deepa,AP,Chandigarh Engineering College

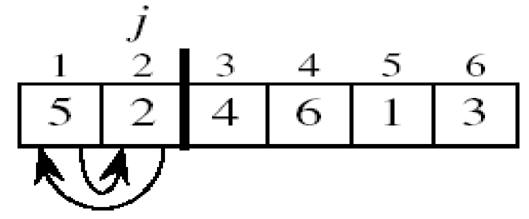
Proving Loop Invariants

- Proving loop invariants works like induction
- Initialization (base case):
  - It is true prior to the first iteration of the loop
- Maintenance (inductive step):
  - If it is true before an iteration of the loop, it remains true before the next iteration
- Termination:
  - When the loop terminates, the invariant gives us a useful property that helps show that the algorithm is correct
  - Stop the induction when the loop terminates

Loop Invariant for Insertion Sort

- Initialization:
  - Just before the first iteration, j = 2:

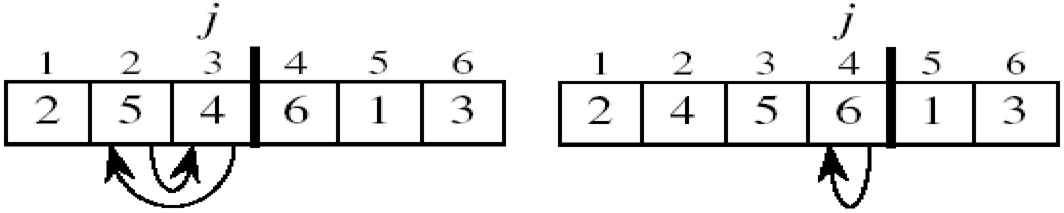
    the subarray A[1 . . j-1] = A[1], (the element originally in A[1]) – is sorted



By Dr.A.Deepa,AP,Chandigarh Engineering College

Loop Invariant for Insertion Sort

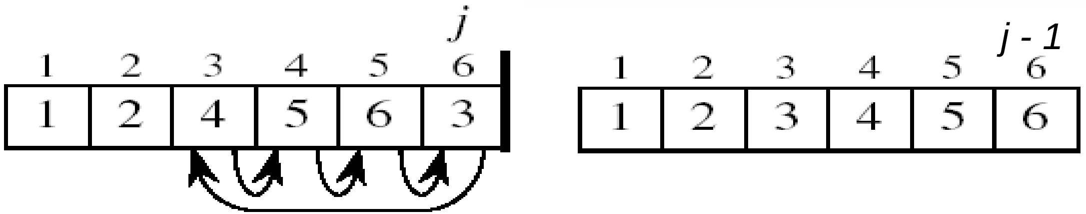
- Maintenance:
  - the while inner loop moves A[j -1], A[j -2], A[j -3], and so on, by one position to the right until the proper position for key (which has the value that started out in A[j]) is found
  - At that point, the value of key is placed into this position.



By Dr.A.Deepa,AP,Chandigarh Engineering College

Loop Invariant for Insertion Sort

- Termination:
  - The outer for loop ends when j = n + 1 ⇒ j-1 = n
  - Replace n with j-1 in the loop invariant:
    - the subarray A[1 . . n] consists of the elements originally in A[1 . . n], but in sorted order



- The entire array is sorted!

Invariant: at the start of the for loop the elements in A[1 . . j-1] are in sorted order

By Dr.A.Deepa,AP,Chandigarh Engineering College

Insertion Sort - Summary

- Advantages
  - Good running time for "almost sorted" arrays $\Theta(n)$
- Disadvantages
  - $\Theta(n^2)$ running time in worst and average case
  - $\approx n^2/2$ comparisons and exchanges

By Dr.A.Deepa,AP,Chandigarh Engineering College

Thank You

Queries????