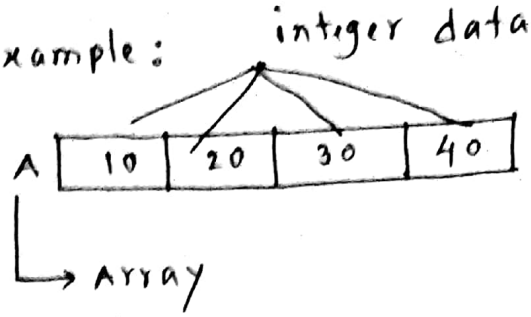
Data structures:

A particular way of organizing data in a computer so that it can be used effectively.

For example:



We can store a list of items having the same data-type using the array data structure.

Different Data structures

- Linear: arrays, lists
- Tree: Binary, heaps, space partitioning, etc
- stack
- Queue
- Graph
- Files
- Records

Uses of Data structures

- Operating system
- Compiler design
- Artificial Intelligence
- Graphics

Fundamental data structures

- Array
- Linked list

common data types

Integer, Floating point, character, string, Boolean

10 -20 10.0 -20.5 'A' 'b' "Rahim" "123" yes No

What is algorithms?

step by step problem solution Technique

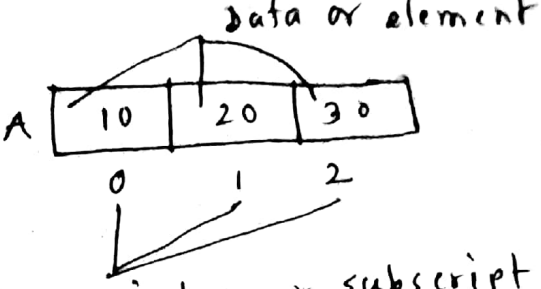
Algorithm Representation

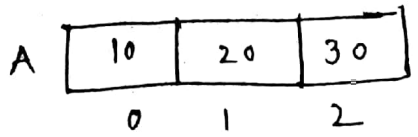
- Pseudocode
- flowchart
- Program

Input a, b
sum = a + b
output sum

Array Data structure

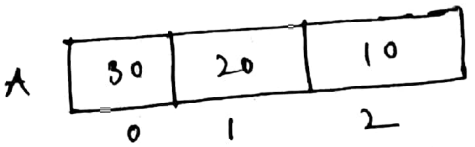
Sorting:



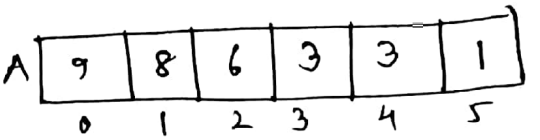


Ascending order

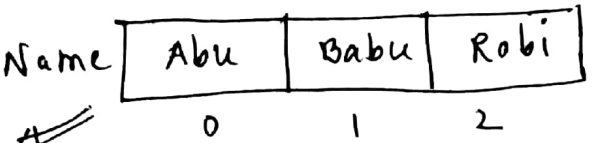
$A[0] = 10$
$A[1] = 20$
$A[2] = 30$



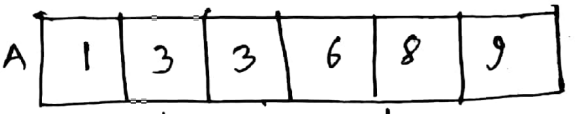
Descending order



Non-increasing order



Alphabetical order



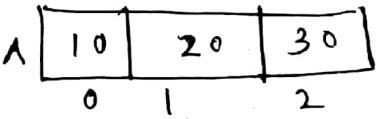
Non-decreasing order

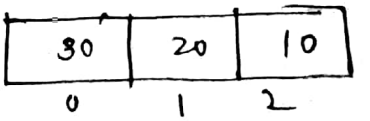
A
B
R

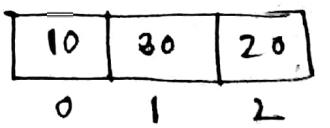
Different types of sorting Techniques

- Bubble sort
- Selection sort
- Merge sort *
- Insertion sort *
- Quick sort *
- Heap sort *
- Linear time sorting
  - Counting sort *
  - Radix sort *
  - Bucket sort

Different cases of sorting: (sort in ascending order)

Best case: 

Wort case: A 

Average case: A 

Counting Sort

Sort the Array A in ascending order using counting sort

	1	2	3	4	5	6
A	4	2	3	2	4	1

Ans:

Input Data: A

	1	2	3	4	5	6
A	4	2	3	2	4	1

Find Max number: $k = 4$

Initialize Mapping Array: C

	0	1	2	3	4=k
C	0	0	0	0	0

count number stored in Array A:

	0	1	2	3	4
C	0	1	2	1	2

Find Proper Mapping Array:

	0	1	2	3	4
C	0	1	3	4	6

output

	0	1	2	3	4	5	6
	1	2	2	3	4	4	

$j=6$ $A[6]=1$ $c[1]=1$ $B[1]=1$ $c[1]=0$

B	1					
	1	2	3	4	5	6

C	0	0	3	4	6
	0	1	2	3	4

$j=5$ $A[5]=4$ $c[4]=6$ $B[6]=4$ $c[4]=5$

B	1					4
	1	2	3	4	5	6

C	0	0	3	4	5
	0	1	2	3	4

$j=4$ $A[4]=2$ $c[2]=3$ $B[3]=2$ $c[2]=2$

B	1		2			4
	1	2	3	4	5	6

C	0	0	2	4	5
	0	1	2	3	4

$j=3$ $A[3]=3$ $c[3]=4$ $B[4]=3$ $c[3]=3$

B	1		2	3		4
	1	2	3	4	5	6

C	0	0	2	3	5
	0	1	2	3	4

$j=2$ $A[2]=2$ $c[2]=2$ $B[2]=2$ $c[2]=1$

B	1	2	2	3		4
	1	2	3	4	5	6

C	0	0	1	3	5
	0	1	2	3	4

$j=1$ $A[1]=4$ $c[4]=5$ $B[5]=4$ $c[4]=4$

B	1	2	2	3	4	4
	1	2	3	4	5	6

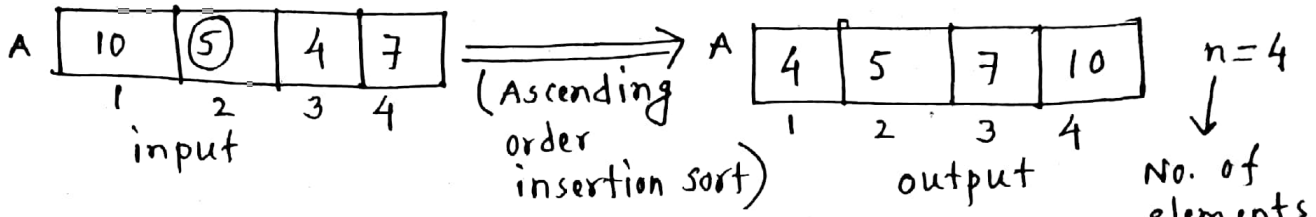
C	0	0	1	3	4
	0	1	2	3	4

Algorithm

```
countingSort(A, B, k) {
    for i = 0 to k
        c[i] = 0
    for j = 1 to n
        c[A[j]] = c[A[j]] + 1
    for i = 2 to k
        c[i] = c[i] + c[i-1]
    for j = n downto 1
        B[c[A[j]]] = A[j]
        c[A[j]] = c[A[j]] - 1
}
```

Insertion Sort

Show the mechanism of insertion sort for the following data



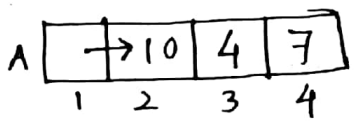
Ans:

$j=2$ $t=A[2]=5$ $i=j-1=2-1=1$

$i>=1$ AND $A[i]>t$

$1>=1$ AND $A[1]>5$

$\Rightarrow$ True AND $10>5$ (True) = True

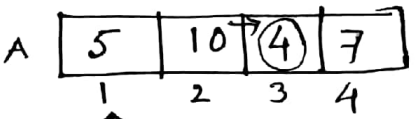


$i=0$

$0>=1$ AND —

$\Rightarrow$ False AND — = False

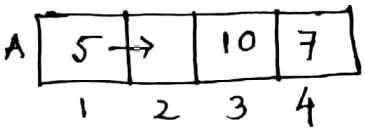
$i+1=0+1=1$



$j=3$ $t=A[3]=4$ $i=j-1=3-1=2$

$2>=1$ AND $A[2]>4$

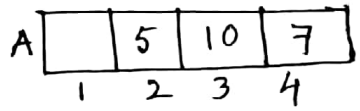
$\Rightarrow$ True AND $10>4$ = True



$i=1$

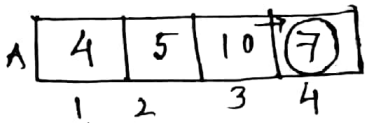
$1>=1$ AND $A[1]>4$

$\Rightarrow$ True AND $5>4$ = True



$i=0$

$0>=1$ AND — = False

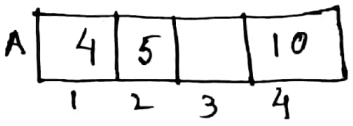


$i+1=0+1=1$

$j=4$ $t=A[4]=7$ $i=j-1=4-1=3$

$3>=1$ AND $A[3]>7$

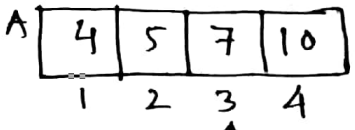
$\Rightarrow$ True AND $10>7$ = True



$i=2$

$2>=1$ AND $A[2]>7$

$\Rightarrow$ True AND $5>7$ = False

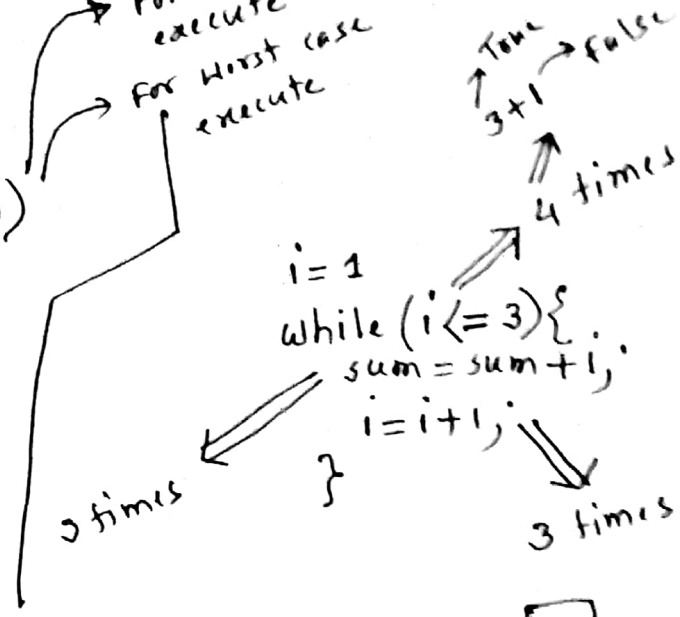


$i+1=2+1=3$

Algorithm

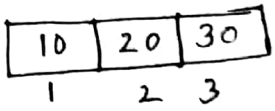
```
L1: for j = 2 to n
L2:     t = A[j]
L3:     i = j-1
L4:     while ((i>=1) AND (A[i]>t))
L5:         A[i+1] = A[i]
L6:         i=i-1
        end while
L7:     A[i+1] = t
    end for
```

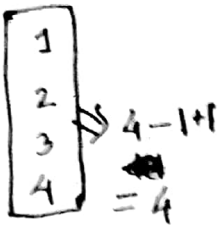
For best case execute one time
For Worst case execute



Time Complexity Analysis

Best case Analysis:





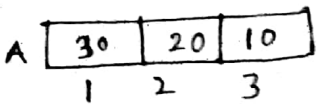
Line Number	Iteration	Time/Iteration	Time
L1	$(n-2+1)+1$ (True / False)	c_1	$c_1 n$
L2	$n-1$	c_2	$c_2(n-1)$
L3	$n-1$	c_3	$c_3(n-1)$
L4	$n-1$	c_4	$c_4(n-1)$
L5	×	×	×
L6	×	×	×
L7	$n-1$	c_7	$c_7(n-1)$

Total running time, $T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4(n-1) + c_7(n-1)$

$= (c_1 + c_2 + c_3 + c_4 + c_7)n + (-c_2 - c_3 - c_4 - c_7)$

$= An + B = O(n)$

Worst Case Analysis



$j=2$ $t=20$ $i=1$ True, $i=0$ False } 2 times ⇓ j

$j=3$ $t=10$ $i=2$ True, $i=1$ True, $i=0$ False } 3 times ⇓ j times

Line Number	Iteration	Time/Iteration	Time
L1	$\underbrace{(n-2+1)}_{\text{True}} + \underbrace{1}_{\text{False}} = n$	c_1	$c_1 n$
L2	$n-1$	c_2	$c_2(n-1)$
L3	$n-1$	c_3	$c_3(n-1)$
L4	$\sum_{j=2}^{n} j$	c_4	$c_4 \sum_{j=2}^{n} j$
L5	$\sum_{j=2}^{n} (j-1)$	c_5	$c_5 \sum_{j=2}^{n} (j-1)$
L6	$\sum_{j=2}^{n} (j-1)$	c_6	$c_6 \sum_{j=2}^{n} (j-1)$
L7	$n-1$	c_7	$c_7(n-1)$

Total running time, $T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \sum_{j=2}^{n} j + c_5 \sum_{j=2}^{n} (j-1) + c_6 \sum_{j=2}^{n} (j-1) + c_7(n-1)$

$= c_1 n + c_2(n-1) + c_3(n-1) + c_4(2+3+\cdots+n) + c_5(1+2+\cdots+\overline{n-1}) + c_6(1+2+3+\cdots+\overline{n-1}) + c_7(n-1)$

$= c_1 n + c_2(n-1) + c_3(n-1) + c_4(\overline{1+2+3+\cdots+n} - 1) + c_5 \frac{n(n-1)}{2} + c_6 \frac{n(n-1)}{2} + c_7(n-1)$

$1+2+\cdots+n = \frac{n(n+1)}{2}$

$1+2+\cdots+(n-1) = \frac{(n-1)n}{2}$

$$= c_1 n + c_2(n-1) + c_3(n-1) + c_4\left\{\frac{n(n+1)}{2} - 1\right\} + c_5\frac{n(n-1)}{2} + c_6\frac{n(n-1)}{2} + c_7(n-1)$$

$$= c_1 n + c_2 n - c_2 + c_3 n - c_3 + \frac{c_4}{2}n^2 + \frac{c_4}{2}n - c_4 + \frac{c_5}{2}n^2 - \frac{c_5}{2}n + \frac{c_6 n^2}{2} - \frac{c_6}{2}n + c_7 n - c_7$$

$$= \left(\frac{c_4}{2} + \frac{c_5}{2} + \frac{c_6}{2}\right)n^2 + \left(c_1 + c_2 + c_3 + \frac{c_4}{2} - \frac{c_5}{2} - \frac{c_6}{2} + c_7\right)n + (-c_2 - c_3 - c_4 - c_7)$$

$$= An^2 + Bn + C$$

$$= O(n^2)$$

Average case Analysis

condition of while loop $= \frac{\text{best} + \text{Worst}}{2} = \frac{1+j}{2}$

Line Number	Iteration	Time / Iteration	Time
L1	n	c_1	$c_1 n$
L2	$n-1$	c_2	$c_2(n-1)$
L3	$n-1$	c_3	$c_3(n-1)$
L4	$\sum_{j=2}^{n} \frac{1+j}{2}$	c_4	$c_4 \sum_{j=2}^{n} \frac{1+j}{2}$
L5	$\sum_{j=2}^{n} \left\{\frac{1+j}{2} - 1\right\}$	c_5	$c_5 \sum_{j=2}^{n} \frac{j-1}{2}$
L6	$\sum_{j=2}^{n} \left\{\frac{1+j}{2} - 1\right\}$	c_6	$c_6 \sum_{j=2}^{n} \frac{j-1}{2}$
L7	$(n-1)$	c_7	$c_7(n-1)$

Total running time, $T(n) = c_1 n + c_2(n-1) + \cdots + c_7(n-1)$

$\vdots$

$= O(n^2)$