1812h01160 01
Vishwas Acharya

Assignment - 3

Q.1) Explain Divide & Conquer (D&C) technique in detail.

Ans Many useful algorithm are recursive in structure to solve a given problem they call themselves recursively one or more time.

- These algorithm typically follows Divide and conquer approach:
- It involves three steps at each level of recursion:

1. **Divide**: Break the problem into several sub problems which are similar to original problems but small in size.

2. **Conquer**: Solve the sub problem recursively. If the sub problem are small enough in size, just solve it in a straight-foward manner.

3. **Combine**: Combine a solution to create a solution of the original problem.

Running Time Analysis

- The running time analysis of such divide and conquer algorithm are almost automatic.
- let $g(n)$ be the time required by D&C on instance of size n.
- The total time $t(n)$ taken by a D&C algorithm is given by recurrance relation.

$$t(n) = 1t\left(\frac{n}{b}\right) + g(n)$$

- The solution of equation is given as,

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

$$t(n) = \begin{cases} \theta(n^k) & \text{if } 1 < b^k \\ \theta(n^k \log n) & \text{if } 1 = b^k \\ \theta(n^{\log_b 1}) & \text{if } 1 > b^k \end{cases}$$

k is power of n in g(n).

Q.2) Solve this Binary search - Examples.

- Demonstrate binary search algorithm and find the element x = 12 in the following array.

2, 5, 8, 12, 16, 23, 38, 56, 72, 91

Ans

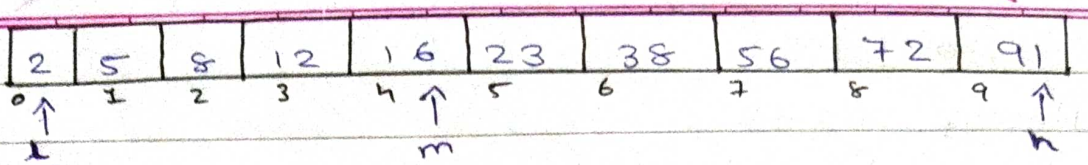
2	5	8	12	16	23	38	56	72	91
0	1	2	3	4	5	6	7	8	9

Here, x = 12

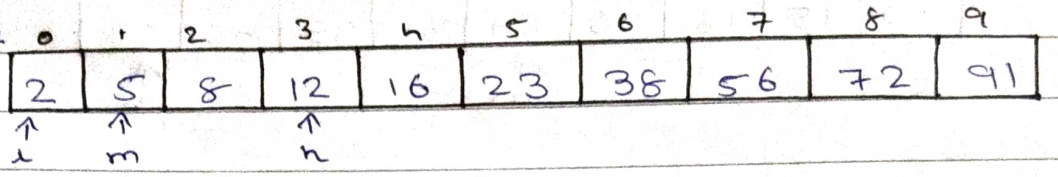
Let take l = low, h = high, m = midpoint (Taking floor value)

l	h	midpoint (m)	
0	9	$\frac{0+9}{2} = 4$	x < m then change h position
0	3	$\frac{0+3}{2} = 1$	x > m then change l position
2	3	$\frac{2+3}{2} = 2$	x > m then change l position
3	3	$\frac{3+3}{2} = 3$	x = m

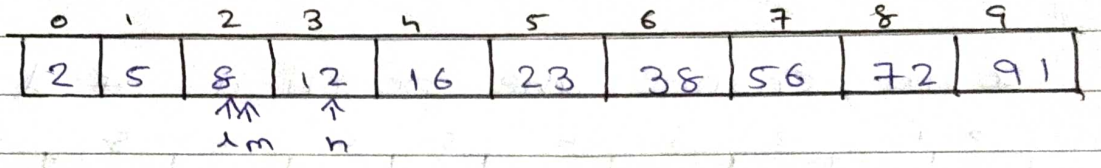
- Hence, we got our required number in four step's at 3rd position.
- For graphical representation, look further



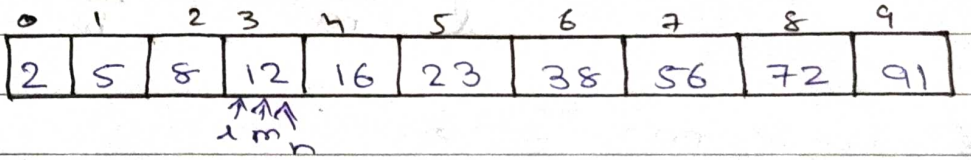
[x < m]



[x > m]



[x > m]



[x = m]

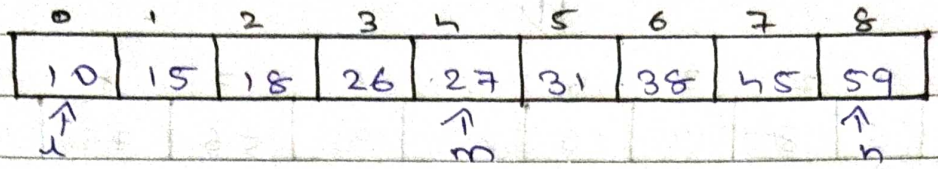
- Explain binary search algorithm and find the element x = 31 in the following array.

  10, 15, 18, 26, 27, 31, 38, 45, 59

Ans - Binary search is an extremely well-known instance of divide-and-conquer approach.
- Let T[1 ... n] be an array of increasing sorted order; that is T[i] ≤ T[j] whenever $1 \le i \le j \le n$.
- Let x be some number. The problem consists of finding x in the array T if it is there.
- If x is not in the array, then we want to find the position where it might be inserted.

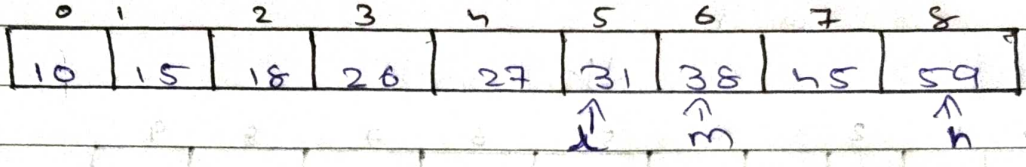
⇒ Here, $x = 31$

Let l = low, h = High, m = midpoint (Taking floor value)



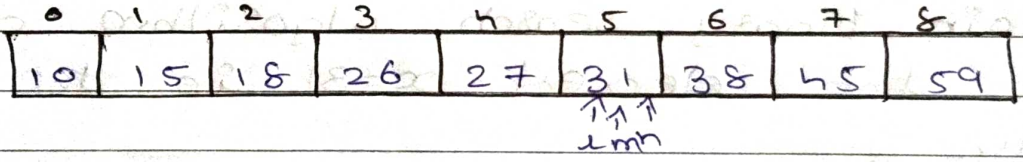
$l = 0$, $h = 8$, $m = \frac{0+8}{2} = 4$

$x > m$, change the position of l



$l = 5$, $h = 8$, $m = \frac{5+8}{2} = 6$

$x < m$, change the position of h



$l = 5$, $h = 5$, $m = \frac{5+5}{2} = 5$

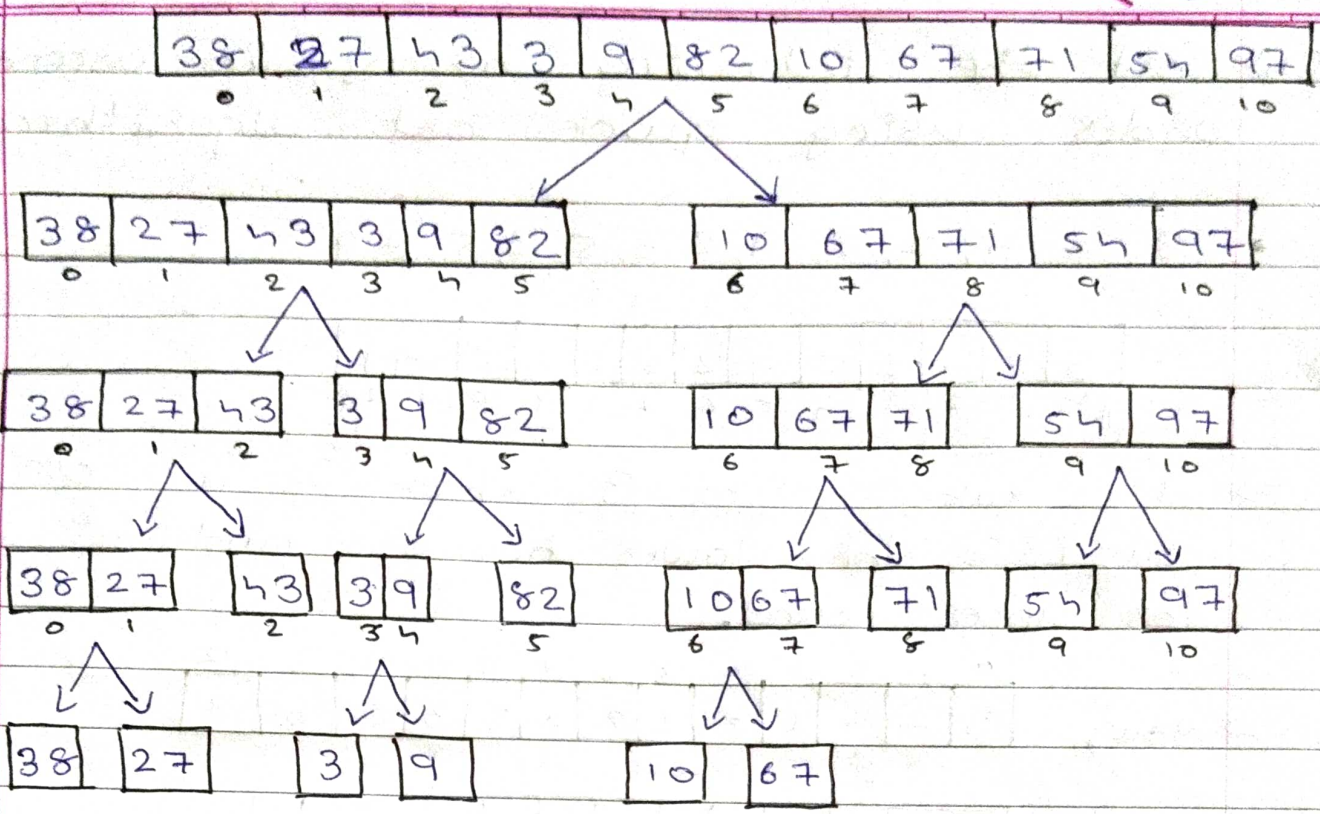
$x = m$,

Hence, we got our requised numbers, at 5th position.

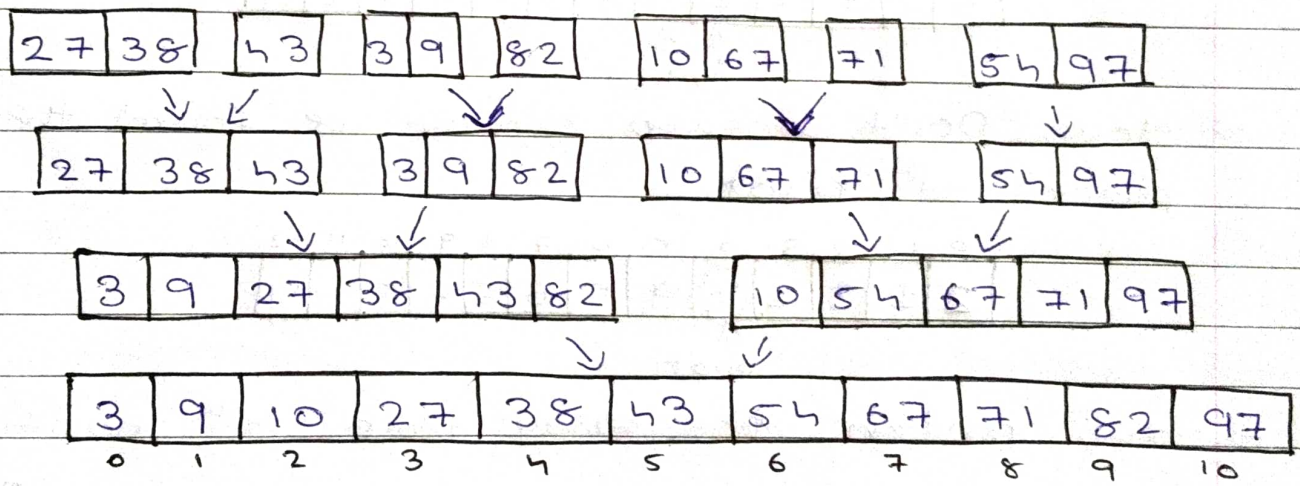
Q.3) Sort given numbers into descending order using merge sort.

38, 27, 43, 3, 9, 82, 10, 67, 71, 54, 97

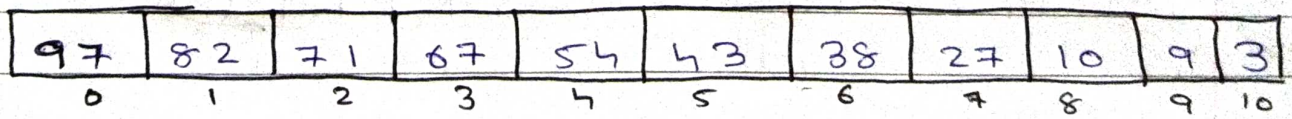
Ans



Merge



=> Above ~~give~~ sorted assay is in ascending order, so for descending order just reverse the assay.

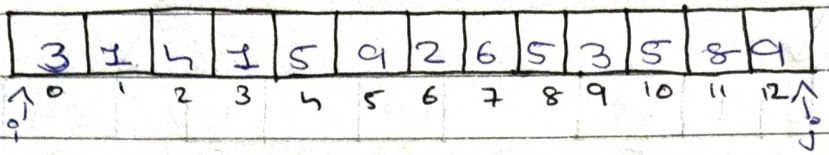


=> Hence, sorted

Q.4) Sort the following array in ascending order using quick sort algorithm.

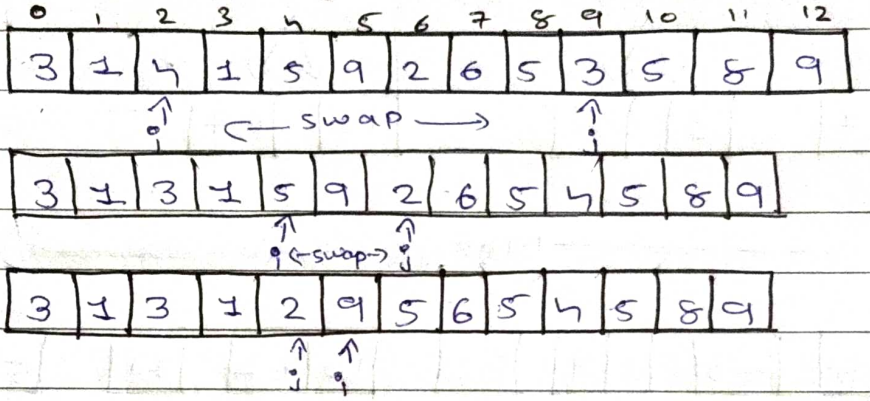
- 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9

Ans

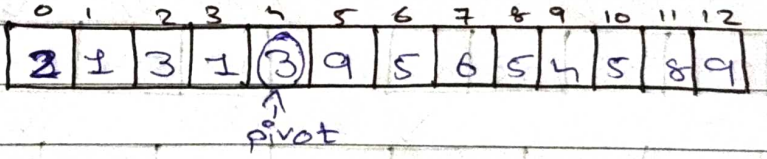


=> Let's take pivot = 3 which is a 0th place.
- Now, i is for lower bound and j is for upper bound.

=> Now,

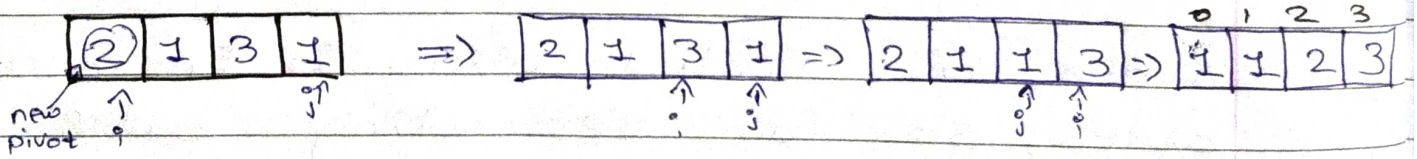


=> Now, Don't swap because we found the right place of pivot.

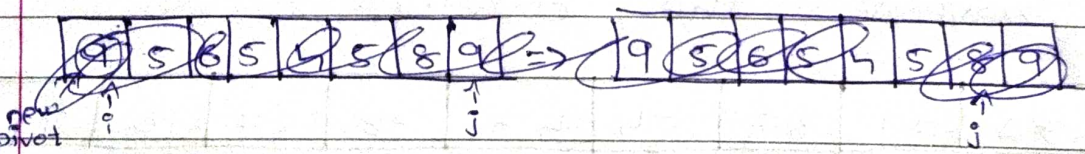


=> Now, portion the array from 4th position.

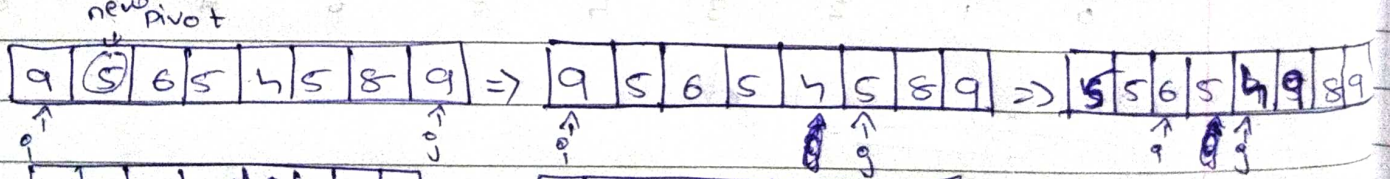
left side

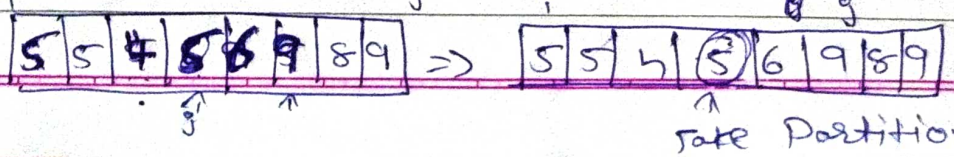


Right side

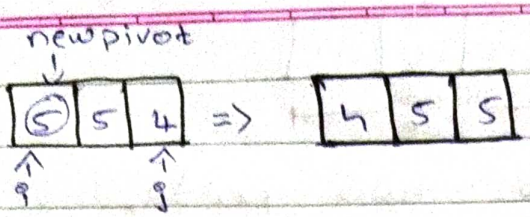


Right side

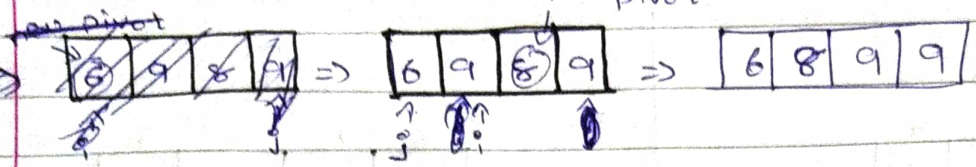




Left side =>

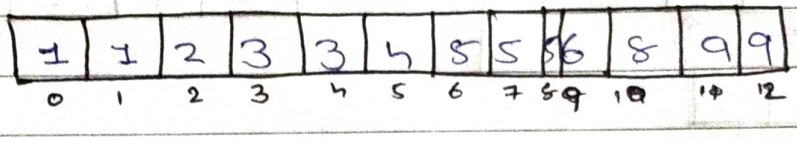


Right side =>



=> Now, combine all the final answers

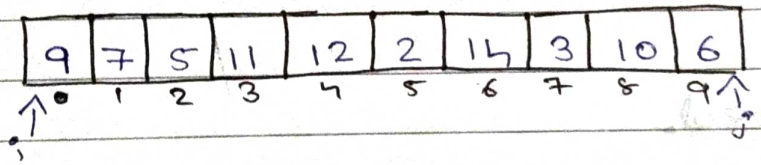
=> we get,



=> Hence, sorted.

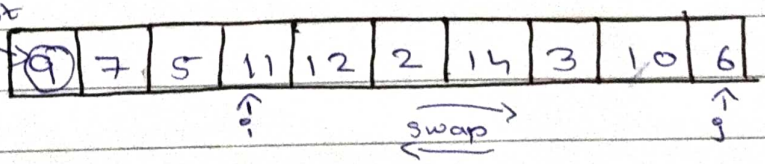
- 9, 7, 5, 11, 12, 2, 14, 3, 10, 6

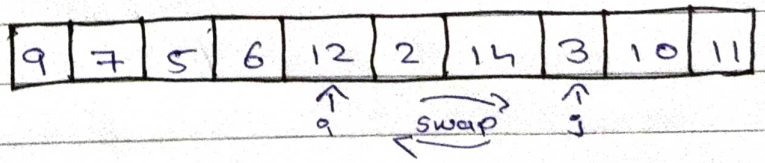
Ans

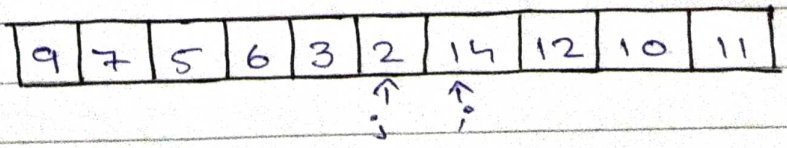


-> Here, pivot = 9 , i = lower bound , j = upper bound

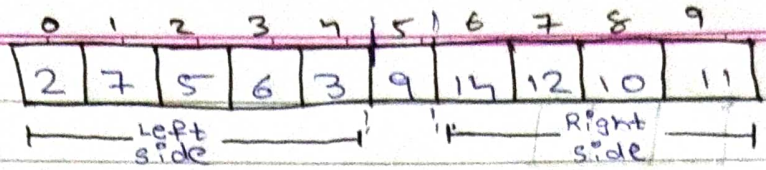
=>





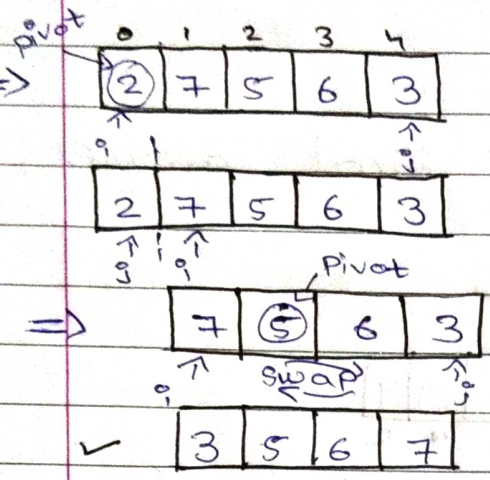


-> Stop here and put pivot on to j's place, and replace j's position value with it.

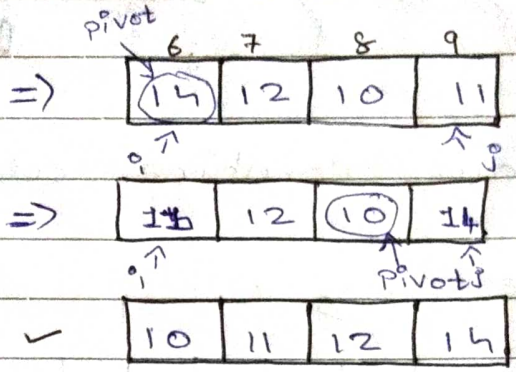


⇒ Divide above array into two parts.

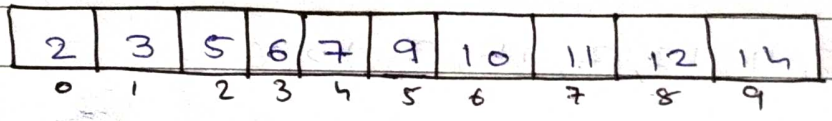
Left side



Right side



⇒ Now, combine all the sorted arrays.



⇒ Hence, sorted.