# DSA Tutorial Question
## Tutorial - 03

1RN21MC071
Pinku Maity

Q. i) Apply the suitable searching Techniques for the following data.

a) 6, 16, 4, 9, 23, 45 key - 50 [Searching element]

→ Pseudocode for linear search since the array is not in sorted format.

```
int linearsearch (int a[], int n, int key)
{
  int i;
  for (i=0; i<n; i++)
  {
    if (a[i] == key)
      return i
  }
  return (-1);
}
```

a [6 | 16 | 4 | 9 | 23 | 45] → Given Array

key = 50

First it will check a[i] == key

on a[i] is a[0] ≠ key [50]

Hence, Next,

a[i] will get incremented, it will Became 1.

| i | n | a[i] element |
|---|---|---|
| 0 | 6 | a[0] 6 ≠ key i.e 50 |
| 1 | 6 | a[1] 16 ≠ 50 |
| 2 | 6 | a[2] 4 ≠ 50 |
| 3 | 6 | a[3] 9 ≠ 50 |
| 4 | 6 | a[4] 23 ≠ 50 |
| 5 | 6 | a[5] 45 ≠ 50 |

∴ Hence return -1.

No element matches the key element.

1

ii) 2, 4, 15, 23, 45

The array is already in Sorted format. Hence Binary Search technique can be applied.

Pseudo Code :

```
int BinarySearch(int a[], int n, int key)
{
int low = 0, high = n-1, mid ;
while (low <= high)
{
  mid = (low + high)/2;
  if (key == a[mid])
     return mid.
  elseif (key < a[mid])
     high = mid - 1;
  else
     low = mid + 1;
}
return (-1)
}
```

→ Given Array

| 2 | 4 | 15 | 23 | 45 |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |

low = 0, high

if low <= high, yes

0 ≤ 4

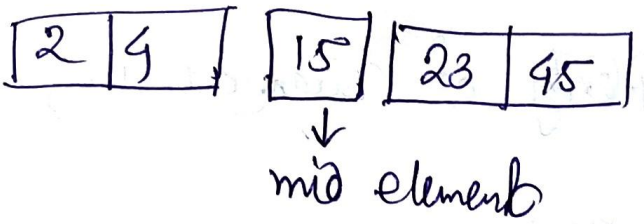
Therefore,

mid = (low + high)/2

i.e (0+4)/2

= 4/2

= 2

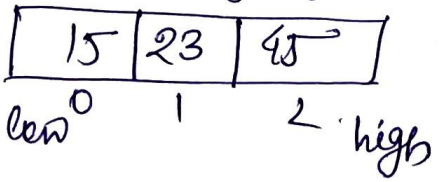



Now check if mid element is equal to key element
i.e $15 \neq 45$

Therefore, now check if key element is less than a[mid]
$45 \not< 15$

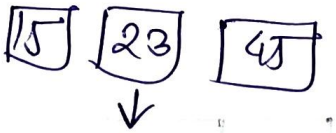
Therefore now the low will be set to
mid+1

i.e, The other half of the array




$mid = (low + high)/2$
$= (0+2)/2$
$= 1$

∴ mid element will be




if key < a[mid]
$45 \not< 23$

∴ Check the other half
∴ $mid = (low + high)/2$
$= 4+4/2$
$= 4$ ∴ The key element is

2. Solve the ~~book~~ worstcase time complexity of quicksort using backward substitution method.

→
$$T(n) = \begin{cases} 0 & \text{if } n=1 \\ T(0)+T(n-1)+n & \text{otherwise} \end{cases}$$

Recurrence relation

$T(n) = T(n-1) + n$

$T(n) = T(n-2) + n + 1$

$T(n) = T(n-4) + (n-3) + (n-2) + T(n-1) + n$

i.e, ...

$1+2+3+ \cdots + n-1+n$

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

$= \frac{n^2+n}{2}$

∴ Worst case is $n^2$.

Replace n with n-1

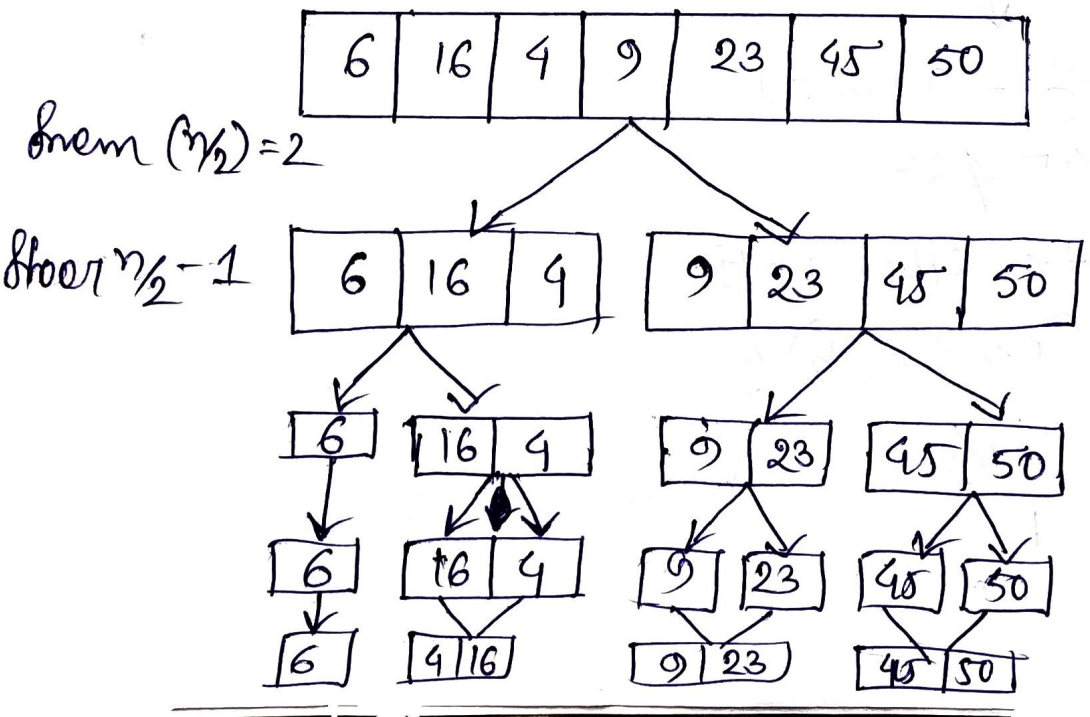
$T(n-1) = T(n-2) + n-1$

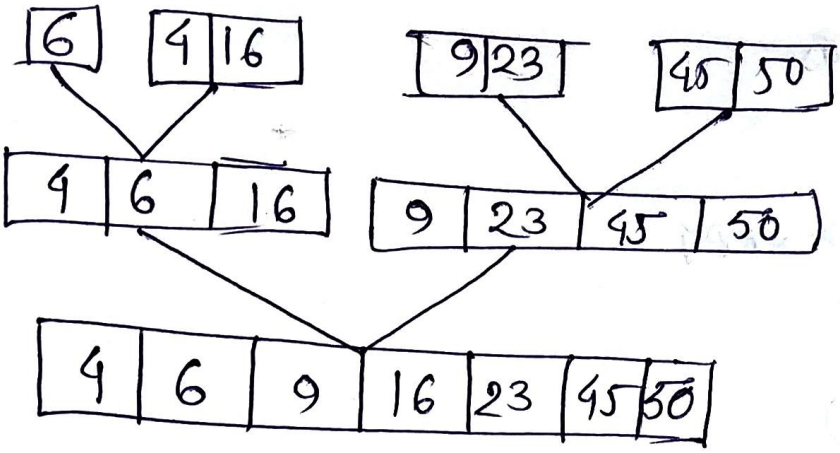
$T(n-2) = T(n-3) + (n-2)$

$= T(n-4) + (n-3)$

3. Apply mergesort for following dataset.

8, 16, 4, 9, 23, 45, 50






---

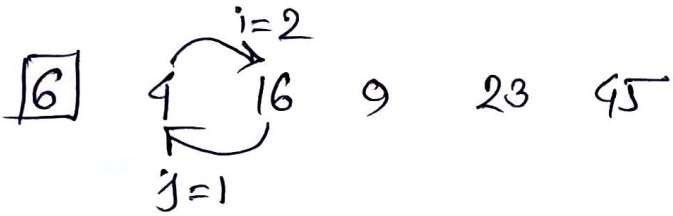
(4) Sort the given data using quick sort.

6, 16, 4, 9, 23, 45

P → be the first element


$j > i$

Swap (A[i], A[j])




$i > j$

Swap (A[i], P)

4 6 16 9 23 45


$j = 3$

$i > j$

Swap (A[j], P)

9 [16] 23 45

[23] j=5
45 → step 4
i=5

i>j

Swap (A[i], P)

23 45

s=4

4 6 9 16 23 45

After sorting.