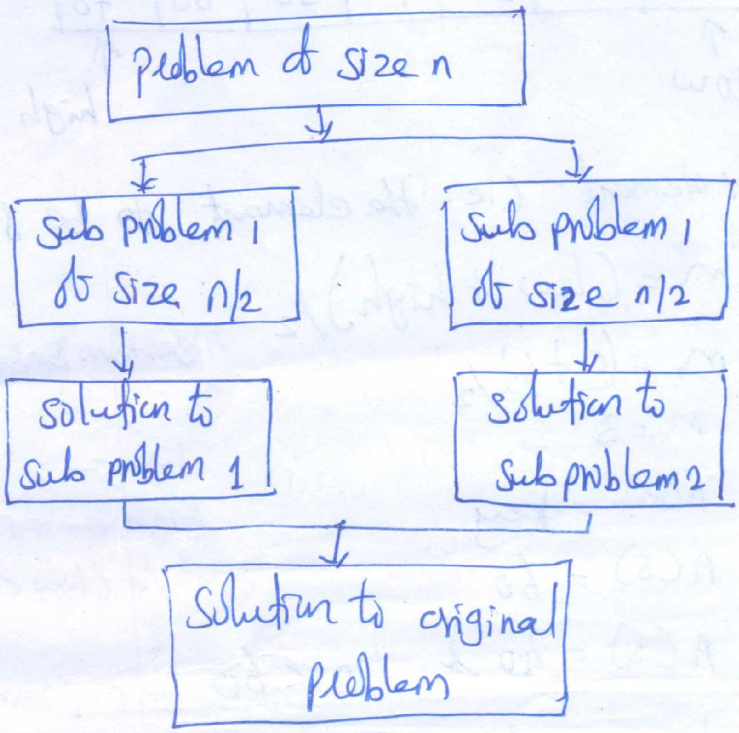
then divide & conquel is reapplied.

- The genelated sub problems are usually of same type as original problem. Hence recussive algorithms are used in divide & conquel strategy.

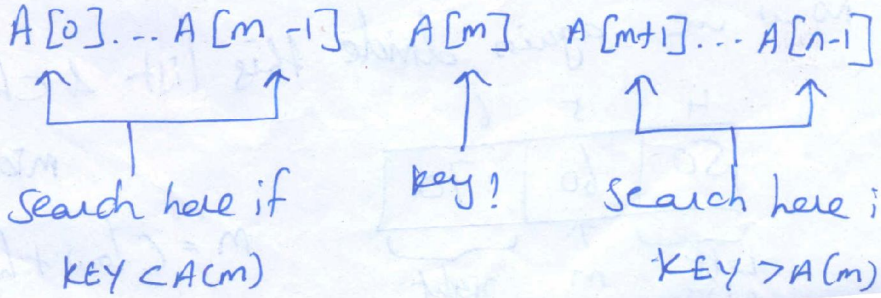


Binary Search:-

Binary Search is an efficient searching method. while searching the elements using this method the most essential thing is that the elements in the array should be sorted one.

A[0 .. n-1] → key element

This can be represented as



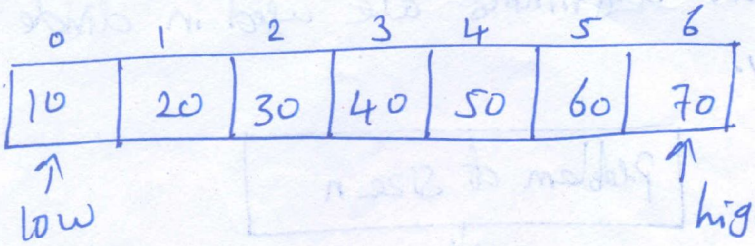
conditions:-

→ otherwise if KEY < A[m] then search the left sub list

→ otherwise if KEY > A[m] then search the right sub list

ex:-

Consider a list of elements sorted in array A as



The KEY element (ie. the element to be searched) is 60.

$m = (low + high)/2$

$m = (0+6)/2$

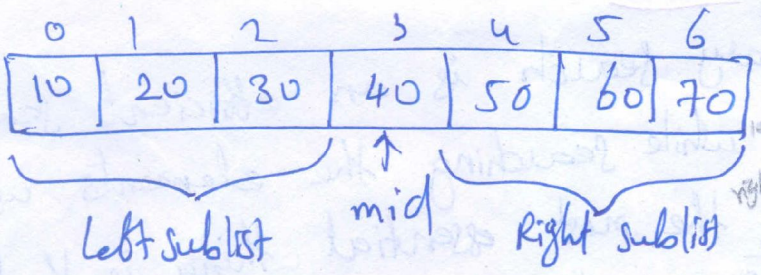
$m = 3$

A(m) = key

A(3) = 60

A(3) = 40 & 40 < 60

∴ search the right sub list

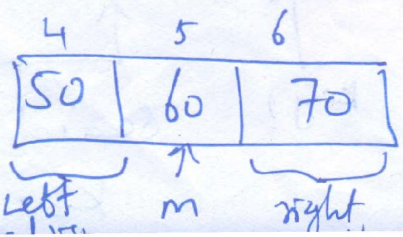


The right sub list is

50	60	70

Now we again divide this list & check the mid element

$m = (low + high)/2$




```
Algorithm BinSearch(A[0..n-1], KEY)
low ← 0
high ← n-1
while (low < high) do
{
    m ← (low + high)/2          // mid value obtain
    if (KEY = A[m]) then
        return m
    else if (KEY < A[m]) then
        high ← m-1              // left
    else
        low ← m+1               // right
}
return -1                       // if element not present in the list
```

$C_{worst}(2^{k-1}) = C_{worst}(2^{k-2}) + 1$

Then ③ becomes

$C_{worst}(2^k) = [C_{worst}(2^{k-2}) + 1] + 1$

$= [C_{worst}(2^{k-2}) + 2]$

Then

$C_{worst}(2^k) = [C_{worst}(2^{k-3}) + 1] + 2$

$= C_{worst}(2^{k-3}) + 3$

$\vdots$

$C_{worst}(2^k) = C_{worst}(2^{k-k}) + k$

$= C_{worst}(2^0) + k$

$= C_{worst}(1) + k$

But eq per eqn ②

$C_{worst}(1) = 1$

$C_{worst}(2^k) = 1 + k$ $\longrightarrow$ $n = 2^k$ taking log (base 2) on both sides

$\log_2 n = \log_2 2^k$

$\log_2 n = k \cdot \log_2 2$

$\log_2 n = k(1)$

$\log_2 2 = 1$

$\therefore C_{worst}(n) = 1 + \log_2 n$

$C_{worst}(n) \approx \log_2 n$ for $n \gg 1$

$\therefore$ Worst case time complexity $= \Theta(\log_2 n)$

We can verify eqn ①

LHS

Above $n = 2^i$

∴ Substitute

$C_{worst}(n) = \log_2 n + 1$

$= \log_2 (2^i) + 1$

$= \log_2 2 + \log_2 2^{i-1} + 1$

$= 1 + \log_2 2^{i-1} + 1$

$= \log_2 2^{i+2}$

LHS = $\log_2 2^{i+2}$

RHS

Above $n = 2^i$

∴ Substitute

$C_{worst}(n/2) + 1$

$= C_{worst}(2^i/2) + 1$

$= C_{worst}(i) + 1$

$A_{Sc\,worst}(n) = \log_2 n + 1$

in same manner

$C_{worst}(i) = (\log_2 i + 1) + 1$

RHS = $\log_2 i + 2$

Avg case

if $n = 1$

only ele 11 is there

$1 \rightarrow [1]$

if $n = 2$ & search key = 22

11	0
22	1

if $n = 4$ & search key = 44

$m = (0+3)/2 = 1$

As $A[1] = 22$ & $22 < 44$

search right sublist,

$m = (2+3)/2 = 2$

$A[2] = 33$ & $33 < 44$

search right sublist

11	0
22	1
33	2
44	3

if $n = 8$ & search key = 88

$m = (0+7)/2 = 3$

$A[3] = 44$ & $44 < 88$, search right sublist

$m = (4+7)/2 = 5$

11	0
22	1
33	2
44	3
55	4
66	5

$m = (7+7)/2 = 7$

$A[7] = 88$, yes ele is located

To summarize above equation

n	Total comparison (c)	
1	1	
2	2	
4	3	above eq
8	4	$\log_2 n + 1 = c$
16	5	if $n = 2$
		$\log_2 2 + 1 = c$
		$1 + 1 = c$
		$c = 2$
		if $n = 8$ then
		$\log_2 8 + 1 = c$
		$3 + 1 = c$
		$\therefore c = 4$

∴ Avg case complexity is $\theta(\log n)$

The basic operation in binary search is comparision of search key with the array elements.

$$C_{worst}(n) = C_{worst}(n/2) + 1 \quad \text{for } n > 1 \quad \text{--- (1)}$$

$C_{worst}(n/2)$: Time required to compare left sublist or right sublist

1: one comparison made with middle element

$$C_{worst}(1) = 1 \quad \text{--- (2)}$$

assume $n = 2^k$

$$C_{worst}(2^k) = C_{worst}(2^k/2) + 1$$

$$C_{worst}(2^k) = C_{worst}(2^{k-1}) + 1 \quad \text{--- (3)} \rightarrow$$

- The worst case time complexity of binary search is $\Theta(\log_2 n)$
- The average case time complexity of binary search is $\Theta(\log n)$. Best case - $\Omega(1)$

Adv of binary search:-

Binary search is an optimal searching algorithm using which we can search the desired element very efficiently.

DisAdv of binary search:-

This algorithm requires the list to be sorted. Then only this method is applicable.

Application:-