```
          sz-1
for i in range(sz):          n
    - - -
✓
```

sz = 6

arr	15	7	22	25	70	17
	0	1	2	3	4	5

n = 70

```
for ( i=0; i<sz; i++)
{     if (arr[i] == n) {
           f=1;
           break}
}
if (f==0)
      printf("%d does not exist";n);

else
      printf("%d exists at %d position of the array", n, i);


if (f==0):
     print(n, "does not exist")

else:
     print(n, "exists at index", i, "of the array")
```

18	25	7	3	75	21	19	70
0	1	2	3	4	5	6	7

Av. Case Complexity $(1+n)/2 \cong n/2$

Binary Search

15	25	40	50	70	90	120
0	1	2	3	4	5	6
				↑ b		↑ e

```
While (beg <= end)
{
  b = 4
  e = 3
  m = 4
  if (arr[mid] == n)
    {
      f=1;
      break;
    }
  elseif (arr[mid] < n)
          beg = mid+1

  else
      end = mid-1
```

beg = 0

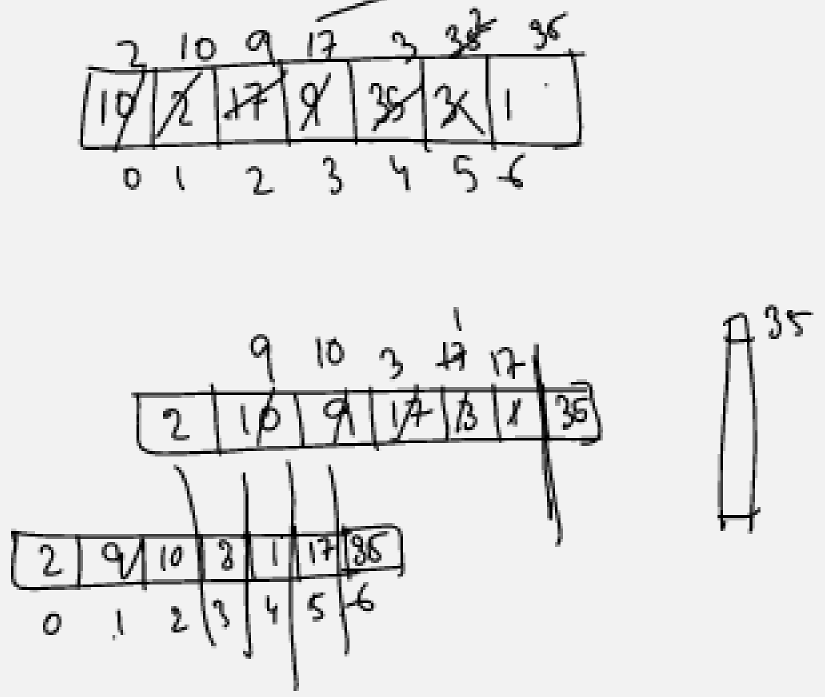
end = sz-1

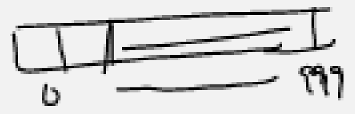
mid = (beg+end)/2

60

$(O) = \log n$

Bubble Sort





```
for i in range(sz):
    for j in range(sz - i - 1):
                  7-1-1
        if (arr[j] > arr[j+1])

            arr[j], arr[j+1] = arr[j+1], arr[j]

            temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp
```

n $\quad n + (n-1) + (n-2) + \cdots \quad 1$

$\sum n$

$2S = n(a+l)$

$S = \frac{n}{2}(a+l)$

$\overbrace{2S = (a+l) + (a+l) + (a+l) \cdots}^{n}$

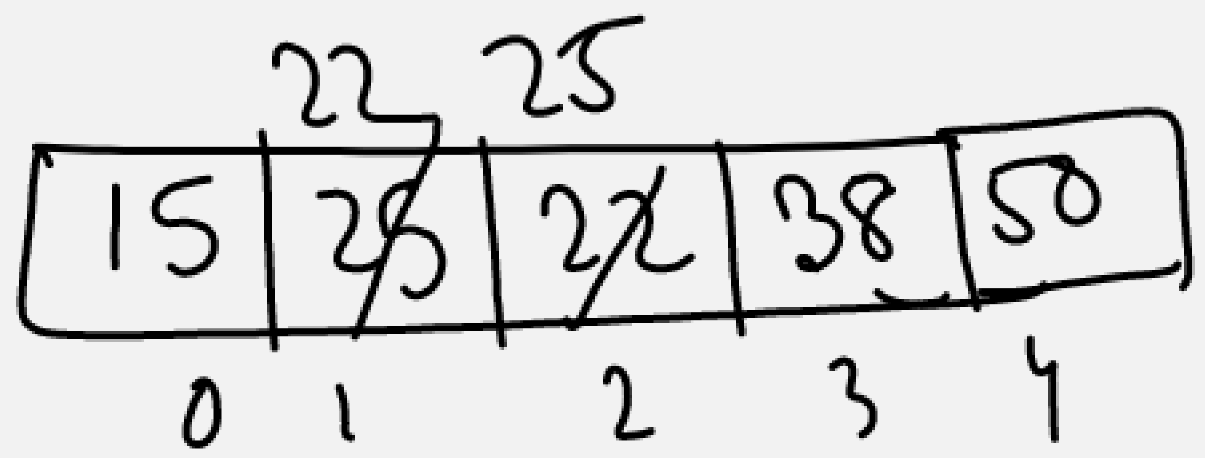
	18	20	40	45	
15	~~20~~	~~18~~	~~45~~	~~40~~	78

15	18	20	40	45	78
0	1	2	3	4	5

```
for  i  in range(sz):
        fl=false
    for j in range(sz-i-1):
            if(lst[j]>lst[j+1]):

                lst[j], lst[j+1] = lst[j+1], lst[j]
                fl= True
    if (fl==False):
        break
```

fl= false



Selection Sort

arr[min]

3	5			12	15	
~~12~~	~~15~~	7	78	~~3~~	~~5~~	30
0	1	2	3	4	5	6
		↑				

4 i=1

min ~~i~~ (~~0~~) 12

~~1~~ ~~2~~ 5 j = (i+1)