

for i in range(^{SZ-1}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) {
        n = 1;
        break;
    }
}
```

if (n == 0)

printf("%d does not exist", n);

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

if (n == 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 $(1+n)/2 \equiv n/2$

Binary Search

while (beg <= end)

{
b = 4

e = 3

m = 4

if (arr[mid] == n)

{
k = 1;
break;

}

else if (arr[mid] < n)

beg = mid + 1

else

end = mid - 1

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

key = 0

end = 52 - 1

↑
6

mid = (beg + end) / 2

↑
e

(60)

$$O() = \log n$$

Bubble Sort

2	10	9	17	3	35	35
10	2	17	9	35	3	1
0	1	2	3	4	5	6

	9	10	3	17	17	
2	10	9	17	3	1	35
0	1	2	3	4	5	6

2	9	10	3	1	17	35
0	1	2	3	4	5	6



for i in range(sz):

for j in range(sz-i-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

$$n + (n-1) + (n-2) + \dots + 1$$

$\sum n$

$$2S = n(a+l)$$

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

$$2S = (a+l) + (a+l) + (a+l) + \dots + n$$

	18	20	40	45	
15	20	18	45	50	78

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

fl = false

for i in range(Sz):

for j in ^{fl = false} 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

	22	25			
15	25	27	38	50	
0	1	2	3	4	

Selection Sort

$$a_{\text{m}} [\text{min}]$$

Handwritten diagram of an array with 7 cells. Above the cells are indices 0 to 6. Above the array are values 3, 5, 12, 15. The array contains values 12, 15, 7, 78, 3, 5, 30. Arrows point from 3 to index 0 and from 5 to index 1. The values 12, 15, 3, and 5 are crossed out with diagonal lines.

$$y_{i=1}$$

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

~~\wedge~~ ~~$2 \leq j = (i+1)$~~