

Searching and Sorting

linear search

arr, target

return the target index if present

else -1

10	23	45	70	11
0	1	2	3	4

, 70

↓

3

run a loop and compare

Binary Search

arr, target
↓
Sorted

0	1	2	3	4	5	6	
10	23	35	45	50	70	85	50

1st Start mid end

1. I will consider the full array/list
start = 0 end = n-1

2. See the middle element.
middle = (end + start) // 2

3. we compare element at middle
with target.

$$\frac{4+4}{2} = 4$$

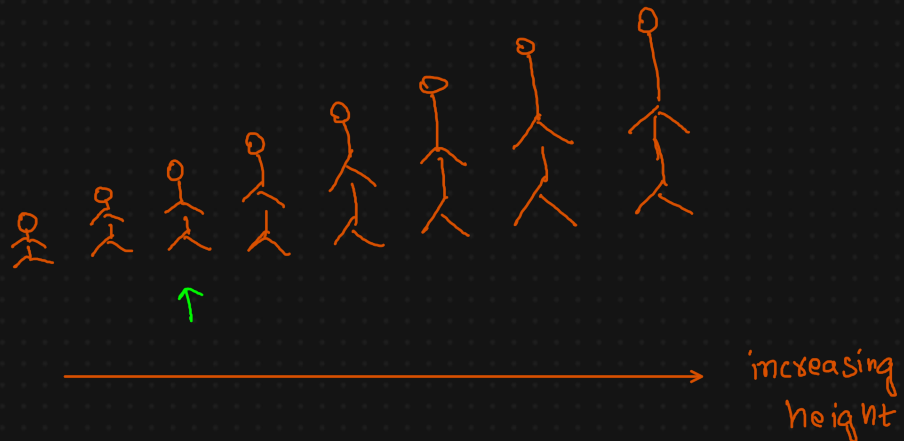
start mid end
4 5 6
↑
start = 4 = mid
end

1st iteration \Rightarrow mid = 3
arr[mid] = 45 < 50

2nd iteration \Rightarrow mid = 5
arr[mid] = 70 > 50

3rd iteration \Rightarrow mid = 4
arr[mid] = 50 == 50

teacher

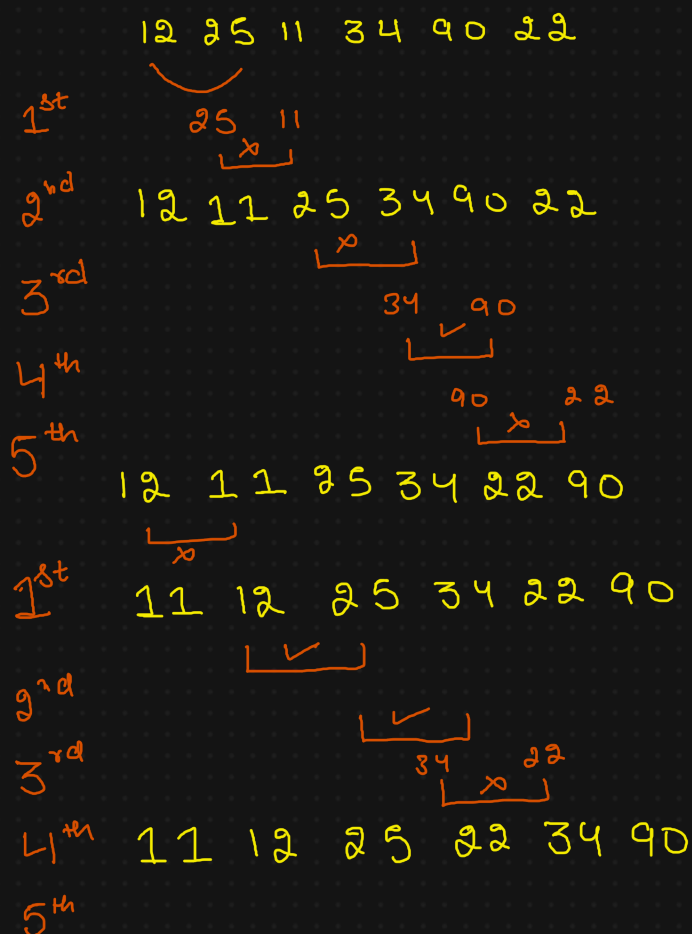


Sorting Algorithm

→ Bubble sort algorithm

1. The largest element i.e. 90 is at the right position
2. We needed $\text{len} - 1$ passes to do that

In 2nd pass, the second largest is at the right position



Bubble Sort Implementation

Pass 1		Pass 2	
j	$j+1$	j	$j+1$
0	1	0	1
1	2	1	2
2	3	2	3
3	4	3	4
4	5	4	5
5	6		

12 25 11 34 90 22

0 1 2 3 4 5

12 11 25 34 22 90

1st pass → $n-1$

2nd pass → $n-2$

3rd pass → $n-3$

range(0, len)

0 — len-1

Selection Sort

Select

In each pass we see the array and select the minimum.

We swap that to the right position

12 25 11 34 90 22



11 12 22 25 34 90

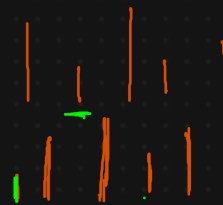
Minimum
Select

$n-1$ passes

Pass 1 11 25 12 34 90 22

Pass 2 11 12 25 34 90 22

Pass 3



Selection Sort Algorithm

$\text{len(arr)} = 6$

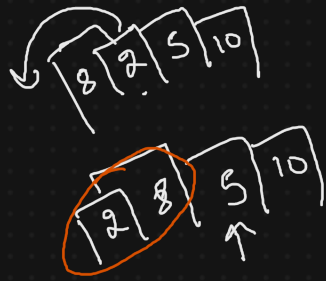
start min end min

0 0 5 \rightarrow 2

12 25 11 34 90 22

0 1 2 3 4 5

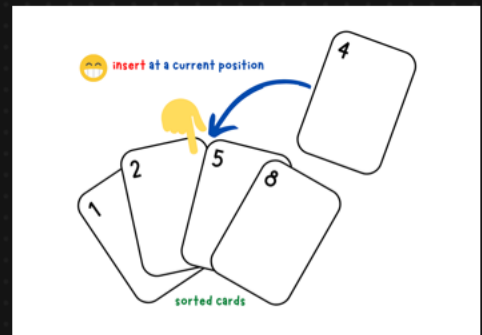
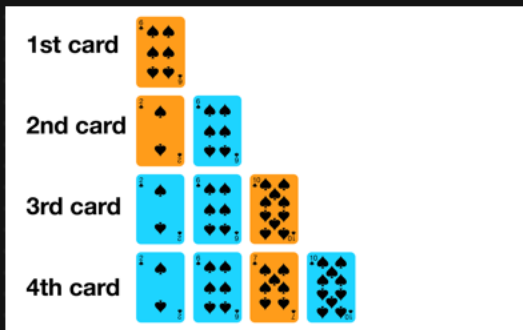
Insertion Sort Algorithm



first round not required
as 1 element always
sorted.

1 2 25 11 34 90 22
0 1 2 3 4 5

Round 1 1 2
Round 2 1 2 25
Round 3 1 1 2 25
Round 4 1 1 2 25 34
Round 5 1 1 2 25 34 90
Round 6 1 1 2 22 25 34 90



Implementation of Insertion sort

{Current card - 1 to 0}

current Card correct position

1 0-0
x

2 1 ✓

current = 11 0 ✓

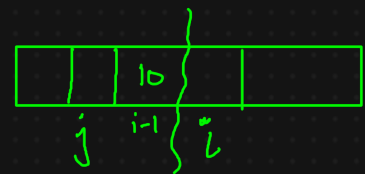
3 2, 1, 0

4 3, 2, 1, 0

5 4
3
2
1
0
current = 22

1 2 25
1 2 25 11 34 90 22
0 1 2 3 4 5
↑

current = 5



j = i-1 ... 0

22 25 34 90
11 12 25 34 90 22
j