LINEAR SEARCH:-

• PSEUDO CODE:-

```
1- def linear search(x,y):          → 1
2- count = 0                        → 1
3- flag = 0                         → 1
4- for i in range (len(x)):         → n+1
5-      count = count+1             → n
6-      If x[i]==y:                 → 1
7-           flag = 1               → 1
8- return flag                      → 1
```

• DRY RUNNING:-

x = [2,3,4,5,6] → worst case when key is not present in list

y = 9

2- count = 0

3- flag = 0

4- 1ST iteration of for:

i = 0

5- count = 1 → 0
2

6- If x[0]==9:

7-

2nd iteration of for:-

4- i = 1

5 count = 2 → 0
3

6- If x[1]==9:

• 3rd iteration of for:

4. i=2

5. count=3

6. If 4==9 :

7. ________

• 4th iteration of for:

4. i=3

5. count=4

6. If 5==9 :

7. ________

• 5th iteration of for:-

4. i=4

5. count=5

6. If 6==9:

7. ________

• 6th iteration of for:-

4. i=5

—x—x— loop terminates

8- return 0

Date

Lines of code	Time/ instruction	frequency (worst case)	$x = [2,3,4,5]$ $y = 2$ (Best case)
1	c	1	1
2	c	1	1
3	c	1	1
4	c	n+1	n+1
5	c	n	n
6	c	n	$\boxed{1}$ → best case mai only 1 comparison hota hai
7	c	0	1
8	c	1	1

$T(n) = c[1+1+1+n+1+n+n+1]$

$T(n) = c[3n+5]$

DISCUSSION:-

- Worst case complexity of linear search is O(n)

- linear search in worst case grows linearly.

Scanned with CamScanner

T(n) AT BEST CASE:-

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

$T(n) = 7+2n$

• Discussion:-

- In best case, search terminates in success with just one comparison
- In best case, complexity of linear search is $O(1)$.
- The element being searched may be found at the first position.