Linear Search: Works on any kind of arrays.
TC: $O(n)$

Binary Search: Can search for a specific element only in sorted arrays.
TC: $O(\log n)$

$\begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 5 & 8 & 10 & 15 & 20 \end{array}$ find = $\boxed{15}$

low = ~~0~~ 4
high = 6

$mid = \frac{low+high}{2} = 3$

$mid = \frac{4+6}{2} = 5$

$\boxed{5}$ ✓

$[O(1)]$

$n \rightarrow \frac{n}{2} \rightarrow \frac{n}{4} \cdots\cdots \rightarrow 1 \Big] \Rightarrow 1 \rightarrow 2 \rightarrow 4 \cdots\cdots \rightarrow \frac{n}{2} \rightarrow n$

first term = 1
mult. factor = 2
⇒ no. of terms = $\underline{k}$

$1 \cdot 2^{k-1} = n \Rightarrow k-1 = \log_2 n$

$\boxed{k = O(\log n)}$

log n

$a \cdot r^{n-1}$

$\begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1, & 3, & 5, & 5, & 5, & 5, & 67, & 123, & 125 \end{array}$, x = 5

first_occ = 2
last_occ = 5

$[2 \Rightarrow O(n)]$

$\begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1, & 3, & 5, & 5, & 5, & 5, & 67, & 123, & 125 \end{array}$ x = 5

low = ~~0~~ 2
high = ~~8~~ ~~3~~ 1

mid = ~~4~~ ~~1~~ 2

first_occ = ~~4~~ ~~1~~ 2

Performing the normal BS but in case arr[mid] == x, we update the first occ & discard the right half.

$mid = \frac{(low + high)}{2}$ $\left[mid = low + \frac{(high - low)}{2} \right]$

$n = 10^5$
$n = 10^{10}$

largest 'mid' s.t. mid * mid ≤ N

$\underline{N = 25}$

low = 0
$high = \frac{25}{2} = 12$

$mid: \frac{0+12}{2} = 6$

4 → 2
~~5~~ → 2
9 → 3

$6 * 6 = \underline{36}$

low = 0
high: 5

$mid: \frac{0+5}{2} = 2$

$2 * 2 = 4$

result = 2

low : 3
high = 5

$mid = \frac{3+5}{2} = 4$

$4 * 4 = 16$

result = 4

low : 5
high = 5

$mid: \frac{5+5}{2} = 5$

$5 * 5 = 25$

$\boxed{result : 5}$

low = 6
high = 5] low > high ⇒ STOP

$\underline{N = 30}$ Find the largest 'mid' s.t. mid² ≤ N

low = 0
high: 15
mid: 7

7 * 7 = 49 > N ⇒ Disc. the right half

low : 0
high: 6
mid = 3

3 * 3 = 9 ≤ N ⇒ result : 3
Disc. the left half

low = 4
high = 6
mid = 5

5 * 5 : 25 ≤ N ⇒ $\boxed{result = 5}$
Disc. the left half

low : 6
high = 6
mid = 6

6 * 6 : 36 > N ⇒ Disc. the right half

low: 6
high = 5] low > high ⇒ Sol.

0 ↑ … 6 ↑

---

n-1
[10, 20, 10, 30] M = 2

Total no. of permutations = $^{n-1}C_m$

$O(^{n-1}C_n \times n)$

We will perform B.S. on the result.

[10, 20, 10, 30] $\underline{M = 2}$

low = 30, high = ~~60~~ 70

[Find the no. of students s.t. no student reads more than 'mid' no. of pages]

(max)
low = 30

(sum)
high = ~~60~~ 70

$mid: \frac{30 + 70}{2} = \cancel{45}\ \underline{50}$

$\boxed{M}$ $\boxed{60 \quad 51, 52}$

$[\underset{1}{10}, \underset{1}{20}, \underset{1}{10}, \underset{2}{30}]$ $\boxed{result = 50}$

⇒ Disc. the right half.

low = 30 high = 49
$mid = \frac{30+49}{2} = \underline{39}$

$[\underset{1}{10}, \underset{1}{20}, \underset{2}{10}, \underset{3}{30}]$
$\underline{3 > M}$

⇒ Disc. the left half.

low: 40 high = 49
$mid = \frac{40+49}{2} = 44$

$[\underset{1}{10}, \underset{1}{20}, \underset{1}{10}, \underset{2}{30}]$
$\boxed{result = 44}$

⇒ Disc. the right half

low = 40 high = 43
$mid = \frac{40+43}{2} = 41$

$[\underset{1}{10}, \underset{1}{20}, \underset{1}{10}, \underset{2}{30}]$
$\boxed{result = 41}$

⇒ Disc. the right half

low = 40 high = 40
$mid = \frac{40+40}{2} = 40$

$[\underset{1}{10}, \underset{1}{20}, \underset{1}{10}, \underset{2}{30}]$
$\boxed{result = 40}$

⇒ Disc. the right half

low = 40 high = 39 ⇒ STOP

⇓
$\underline{result : 40}$

TC: O(log(sum) * n)

0 ↑ — n-1 ↑

⇒ O(log(no. of elements in the initial range))

no. of elements in the initial range = [sum − max element + 1]