Linear Search: $O(n)$ time.

Binary Search → Divide & Conquer

↓

[1, 3, 10, 16, 20, 25, 27]. target = 20

↓

[20, 25, 27].

↓

[20]

$n \to n/2 \to n/4 \to n/8 \to \dots \to 1$

$\log(n)$ iterations

TC = $O(\log n)$.

0	1	2	3	4	5	6
1	3	10	16	20	25	27

t = 20.

low = 4
high = 4
mid = 4

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

[5, 5, 5, 5, 5, 5]. t = 5.

0	1	2	3	4	5	6	7	8
1	3	5	5	5	5	67	123	125

t = 5.

mid = 4 x 2

res = 4 x 2

TC = $O(\log n)$.
AS = $O(1)$

---

sqrt()

n = 30

1 → 30.

i = 1 2 3 4 5 6

res = 1 2 3 4 5

TC = $O(\text{sqrt}(n))$.
AS = $O(1)$.

low = 1 4 6 mid = 15 7 3 5 6
high = 30 15 7 5

res = 5.

TC = $O(\log(n))$
AS = $O(1)$.

$n = 10^6$

$\text{sqrt}(n) = 10^3$ $\log(n) = 6$

n = 30.

f = 5.

[5.0, 5.1, 5.2 . . . . . 5.9]

5.00, 5.01, ... 5.99

low = 0 high = 9.9

$\left(f + \frac{\text{mid}}{100}\right)^2$ → n

0	1	2	3	4	5	6	7	8
5	6	7	8	9	10	1	2	3

t = 2

mid = 4 x 7

→ if LH is sorted:
→ if two lies b/w arr[low ... mid]
→ discard the RH.
→ otherwise
→ discard the LH.
→ otherwise (RH is def. sorted)
→ if two lies b/w arr[mid ... high]
→ discard the LH.
→ otherwise.
→ Discard the RH

TC = $O(\log n)$.
AS = $O(1)$.

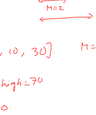
---

books : [10 20 10 30] M = 2.

[10] [20 10 30] → 60
[10 20] [10 30] → 40
[10 20 10] [30]. → 40

low = 30 [M = N].
high = 70 [M = 1].

res :



[10, 20, 10, 30]. M = 2.

low = 70 high = 70.
mid = 50.

↓ LH.

low = 30 high = 47.
mid = $\frac{77}{2}$ = 39.

↓ RH

low = 40 high = 49.
mid = $\frac{89}{2}$ = 44.

↓ LH.

low = 40 high = 43
mid = $\frac{83}{2}$ = 41

↓ LH

low = 40 high = 40.
mid = 40.

↓ LH.

low = 40 high = 39.
STOP!

→ res = 40

10	20	10	30
1	1	1	2

res = 50

10	20	10	30
1	1	2	3

res = 77

10	20	10	30
1	1	1	2

res = 71

10	20	10	30
1	1	1	2

res = 70

TC = $O(n \log(\text{sum}))$

AS = $O(1)$. ~~n~~ k

res = 30
res = 30 30 30 30
30

[10 20 10 30]
mid = 44.

```
bool isPossible(int arr[], int n, int m) {
    int numStudents = 0, numPages = 0;
    for (int i = 0; i < pages.size(); i++) {
        if (numPages + pages[i] <= mid) {
            numPages += pages[i];
        } else {
            ++numStudents;
            numPages = pages[i];
        }
    }

    return numStudent <= m;
}
```