[1, 2, 3, 5, 6, 7, 8, 9, 10]

[2, 3, 5, 6, 7, 8, 9, 10, 1]

[3, 5, 6, 7, 8, 9, 10, 1, 2]

[5, 6, 7, 8, 9, 10, 1, 2, 3] ✓

$O(n\log n + \log n) = O(n\log n)$

$\boxed{O(n)}$

$$\begin{array}{ccccccccc} \downarrow & & & & & \downarrow & & & \downarrow \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ [5, & 6, & 7, & 8, & 9, & 10, & 1, & 2, & 3] \\ & & & & \uparrow & & & & \end{array}$$

Key = 10

$$\times \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ [5, & 1, & 2, & 3, & 4] \\ & & \uparrow & & \end{array}$$

Key = 5

$$\Rightarrow \begin{array}{ccccccccc} \downarrow & & & & & \leftarrow & \downarrow & & \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ [5, & 6, & 7, & 8, & 9, & 10, & 1, & 2, & 3] \\ \cancel{\uparrow} & \cancel{\uparrow} & \cancel{\uparrow} & \cancel{\uparrow} & \cancel{\uparrow} & \cancel{\uparrow} & \cancel{\uparrow} & \cancel{\uparrow} & \uparrow \end{array}$$

[0, 0] ✓ [0, 6] ✗

[0, 1] ✓ [0, 7] ✗

[0, 2] ✓ [0, 8] ✗

[0, 3] ✓

[0, 4] ✓

[0, 5] ✓

$$\text{Key}=7 \quad \begin{array}{ccccccccc} \cancel{\downarrow} & & & & & \downarrow & & & \downarrow \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ [5, & 6, & 7, & 8, & 9, & \boxed{10}, & 1, & 2, & 3] \\ & & & & \cancel{\uparrow} & & \uparrow & & \end{array}$$

$\boxed{\text{Key}=10}$

(9 > arr[0]) ⇒ left half is sorted

Key is greater than the max^m of sorted half

⇒ Key does not occur in the left half.

⇒ move to the right half.

low = 5, high = 8

mid = $\frac{5+8}{2} = 6$

(1 < arr[0]) ⇒ left half is not sorted

⇒ Right half is sorted.

But key does not lies in the range of sorted half

⇒ Discard the sorted half (right half)

⇒ low = 5, high = 5

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

low = 0	high = 6	⇒ mid = 3
low = 0	high = 2	⇒ mid = 1
low = 0	high = 0	

A = [10, 20, 10, 30]

$\boxed{M = 2}$

For M = 4 [10] [20] [10] [30] ⇒ (30)

For M = 1 [10, 20, 10, 30] ⇒ (70)

lower-limit of ans = 30

upper-limit of ans = 70

⇒ Now, since we have a range in which the ans could lie, we will apply the Binary Search on the output (ans).

resut = ~~50~~ ~~44~~ ~~41~~ 40

① low = 30

high = 70

mid = $\frac{30+70}{2} = 50$

Find the no. of Students such that no student reads more than 'mid' no. of pages.

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

⇒ We need 2 students such that none of them reads more than 50 pages.

⇒ Since (2 ≤ M) ⇒ any value greater than or equal to 50 is a valid probable output.

(51, 52 . . . ~~100~~ 60)

⇒ Discard the right half.

② low = 30

high = 49

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

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

⇒ Since $\boxed{3 > M}$, any thing less than or equal to 39 is not a valid probable output.

⇒ Discard the left half.

③ low = 40

high = 49

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

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

⇒ Since (2 ≤ M), anything greater than or equal to 44 are valid probable outputs

⇒ Discard the right half.

④ low = 40

high = 43

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

$[\underset{1}{\underline{10}}, \underset{1}{\underline{20}}, \underset{1}{\underline{10}}, \underset{2}{\underline{30}}]$ ✓

⇒ Anything ≥ 41 is a valid prob. output.

⇒ Discard the right half.

⑤ low = 40

high = 40

mid = 40

$[\underset{1}{\underline{10}}, \underset{1}{\underline{20}}, \underset{1}{\underline{10}}, \underset{2}{\underline{30}}]$ ✓

⇒ Anything ≥ 40 is a valid prob. o/p

⇒ Discard the right half.

⑥ low = 40

high = 39

⇒ Range exhausted

⇒ return the $\boxed{\text{result} = 40}$

$\boxed{\text{TRY : Painter's Partition Problem}}$

$$\begin{array}{cccc} \cancel{\downarrow} & \cancel{\downarrow} & \downarrow & \\ [\underset{1}{\underline{10}}, & \underset{1}{\underline{20}}, & \underset{1}{\underline{10}}, & \underset{2}{\underline{30}}] \end{array}$$

M = 2 mid = 50

student_count = ~~1~~ 2

pages_count = ~~10~~ ~~30~~ ~~40~~ 30

$\boxed{O(\log(\text{sum}))}$

~~low = 1~~ ~~high =~~

high = sum

(sum - low)

No. of B.S. iterations = $O(\log(\text{sum}))$

work done in each B.S. iteration = $O(n)$

⇒ TC = $O(n * \log(\text{sum}))$