

$$\text{mid} = \frac{i+j}{2} = i + \left(\frac{j-i}{2} \right) = \frac{2i + j - i}{2} = \frac{i+j}{2}$$

$$\text{mid} = i + \left(\frac{j-i}{2} \right)$$

69. Sqrt(x)

Easy 7991 4466 Add to List Share

Given a non-negative integer x , return the square root of x rounded down to the nearest integer. The returned integer should be **non-negative** as well.

You **must not use** any built-in exponent function or operator.

- For example, do not use `pow(x, 0.5)` in c++ or `x ** 0.5` in python.

Example 1:

Input: $x = 4$

Output: 2

Explanation: The square root of 4 is 2, so we return 2.

Example 2:

Input: $x = 8$

Output: 2

Explanation: The square root of 8 is

$$x = 4$$

$$\sqrt{4} = 2$$

$$x = 8$$

$$\sqrt{8} = 2.828 = 2$$

$$x = 16$$

$$\sqrt{16} = 4$$

$$x = 17$$

$$\sqrt{17} = 4.123 = 4$$

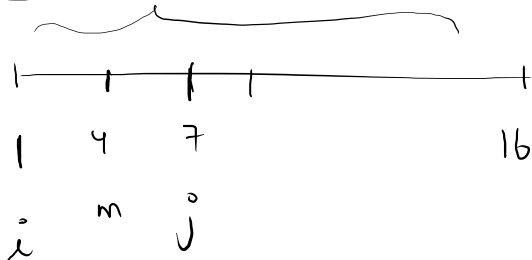
$$x = 16$$

$$\sqrt{x}$$

$$x = 16$$

$$\sqrt{x}$$

answer



} binary search.

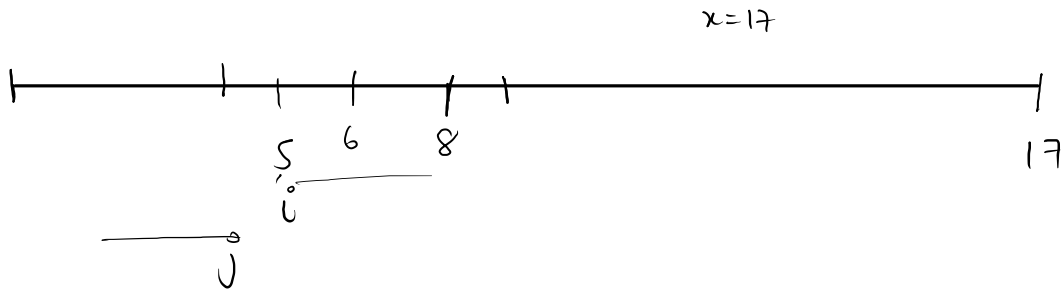
$$m * m == x \rightarrow (m)$$

$$4 * 4 == 16$$

$$\underline{\underline{m * m}} > (x)$$

↳ left

$$m = 9$$



$$ms = 714$$

$$m * m == x \quad \rightarrow \quad av = m$$

$$\underline{m * m > x} \quad \underline{\text{left}}$$

$$m * m < x \rightarrow \text{right (save)}$$



$$m * m == x$$

$$m = x / m$$

$$x=0$$

```
1  class Solution {
2      public int mySqrt(int x) {
3          int ans = 0;
4          int i = 1;
5          int j = x;
6          while(i <= j){
7              int m = i + (j-i)/2;
8
9              if(m == x / m){
10                 return m;
11             }else if(m > x / m){    //left
12                 j = m-1;
13             }else{ //right
14                 ans = m;
15                 i = m + 1;
16             }
17         }
18         return ans;
19     }
20 }
```

The banana challenge

Koko is fond of consuming bananas and is faced with n piles of bananas, where the i th pile has $piles[i]$ bananas. Meanwhile, the guards have temporarily left and are expected to return in h hours.

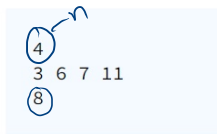
Koko has the freedom to determine her banana-eating speed per hour, which she can set to k . Every hour, she selects a pile of bananas and consumes k bananas from that pile. However, if the selected pile has less than k bananas, she finishes all the bananas in that pile and won't eat any more bananas in that hour.

Koko prefers to eat slowly but is still determined to finish consuming all the bananas before the guards come back.

Return the minimum integer k such that she can eat all the bananas within h hours.

banana eating speed per
hour = k

Sample Input 0



$h=8$



3	6	7	11
6	1	2	3

1 2 3 4 = 10

Sample Output 0

4

$k=3$

10
9
8
7
6

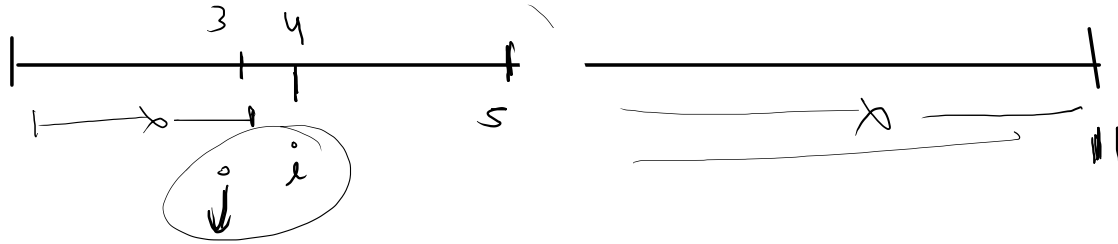
$$h \geq 4$$

$$h = 8$$

3	6	7	11
---	---	---	----

$$\begin{matrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 2 & 3 \end{matrix} = 8$$

$$K = m = 6$$



$$m = 6$$

$$m = 3$$

$$m = 4$$

$$ans = 4$$

```

8 public static void main(String[] args) {
9     Scanner scn = new Scanner(System.in);
10    int n = scn.nextInt();
11    int [] A = new int[n];
12    int max = Integer.MIN_VALUE;
13    for(int i = 0; i < n; i++){
14        A[i] = scn.nextInt();
15        max = Math.max(A[i], max);
16    }
17    int h = scn.nextInt();
18
19    int i = 1;
20    int j = max;
21    int k = max;
22
23    while(i <= j){
24        int m = (i+j)/2;
25        if(isPossible(A, m, h)){
26            k = m;
27            j = m-1;
28        }else{
29            i = m+1;
30        }
31    }
32    return k;
33 }
34 }

```

```

5 public static boolean isPossible(int [] A, int m, int h){
6     ///?? logic
7 }

```

is it possible to eat all banana in
given time.

3 6 7 11

m=6
h=8

```

5 public static boolean isPossible(int [] A, int m, int h){
6     int t = 0;
7     for(int e : A){
8         t += Math.ceil((e*1.0) / m);
9     }
10    return t <= h;
11 }

```


$$h=8$$

$$m=6$$

$$3$$

$$6$$

$$7$$

$$11$$

$$\lceil 3/6 \rceil$$

$$\lceil 6/6 \rceil$$

$$\lceil 7/6 \rceil$$

$$\lceil 11/6 \rceil$$

$$t =$$

$$(1)$$

$$(1)$$

$$(2)$$

$$(2)$$

$$t = 6 \leq h$$

ceiling 10

9.8

floor 9

```

4 public class Solution {
5     public static boolean isPossible(int [] A, int m, int h){
6         int t = 0;
7         for(int e : A){
8             t += Math.ceil((e*1.0) / m);
9         }
10        return t <= h;
11    }

```

$$6 \leq 8$$

$$m=6$$

$$3$$

$$6$$

$$7$$

$$11$$

e

$$t = 0 \times 2 \times 6$$

$$11.0/6 =$$

$$\sqrt{1.}$$

Search Insert position.

Sample Input 1

```
4
1 3 5 6
2
```

$x=2$

1	3	5	6
0	= 1	2	3

Sample Output 1

1

Sample Input 2

```
4
1 3 5 6
7
```

$x=7$

1	3	5	6	7
0	1	2	3	4

Sample Output 2

4

$$n=5$$

$$m=2$$

1	3	5	7
0	1	2	3

$$x=2$$

$$as = \cancel{1} 0$$

$$m = 1$$

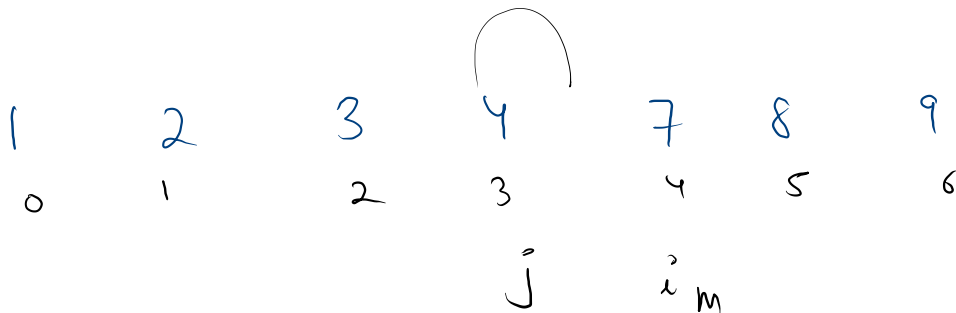
1	2	3	5	7
0	1	1	2	3
j		i		
m				

$$A[m] == x \rightarrow m$$

$$A[m] > x$$

$$A[m] < x$$

$x = 5$
 $\text{ans} = -\cancel{\underline{\underline{3+1}}}$



$$A[m] == x \rightarrow m$$

$$\underline{A[m]} > x \rightarrow \text{left}$$

$$\underline{A[m]} < x \rightarrow \text{save \& right}$$

```
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         int [] A = new int[n];
10        for(int i = 0; i < n; i++){
11            A[i] = scn.nextInt();
12        }
13        int x = scn.nextInt();
14        int ans = -1;
15        int i = 0;
16        int j = n-1;
17        while(i <= j){
18            int m = (i + j)/2;
19            if(A[m] == x){
20                System.out.println(m);
21                return;
22            }
23            else if(A[m] > x){ //left
24                j = m-1;
25            }
26            else{ //A[m] < x -> save and right
27                ans = m;
28                i = m + 1;
29            }
30        }
31
32        System.out.println(ans + 1);
33    }
34 }
```