

Find Square Root

$$n = 25$$

$$\text{ans} = 5$$

$$n = 41$$

$$\text{ans} = 6$$

$$n = 126$$

$$\text{ans} = 11$$

(Advantage :- BS can also be applied on imaginary no.)

Ex:- $n = 30$

apply BS on imaginary series

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30

↑ ↑
j i
 ↑
mid

$$6 \times 6 == 30$$

$$6 \times 6 < 30$$

$$6 \times 6 > 30$$

if ($\text{mid} * \text{mid} == n$) return mid
else if ($\text{mid} * \text{mid} < n$) $i = \text{mid} + 1$
else if ($\text{mid} * \text{mid} > n$) $j = \text{mid} - 1$

$$\text{mid} = \cancel{15} \cancel{7} \cancel{8} \cancel{5} 6$$

$$\text{ans} = j$$

Ex:- $n = 64$

1 7 8 9 10 13 14 27 28 55

↑ ↑

j i

 ↑

 mid

ans = 7

```
loop (i <= j)
    if (mid * mid == n) return mid
    else if (mid * mid < n) i = mid + 1
    else if (mid * mid > n) j = mid - 1;

return j;
```

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    System.out.println(squareRoot(n));
}
public static int squareRoot(int n) {
    int i = 1;
    int j = n;
    while ( i <= j ) {
        int mid = (i + j) / 2;
        if ( mid * mid == n ) {
            return mid;
        } else if ( mid * mid > n ) {
            j = mid - 1;
        } else if ( mid * mid < n ) {
            i = mid + 1;
        }
    }
    return j;
}
```

$$\underline{\underline{T.C = \log(n)}}$$

Search insert position

arr = [⁰1, ¹2, ²5, ³8, ⁴9, ⁵10]

target = 4


insert [⁰1, ¹2, ²4, 5, 8, 9, 10]

template

```
int i = 0, j = n - 1;
while ( i <= j ) {
    int mid = (i + j) / 2;
    if ( arr[mid] == target ) return mid;
    else if ( arr[mid] > target ) j = mid - 1;
    else if ( arr[mid] < target ) i = mid + 1;
}
return i;
```

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int target = scn.nextInt();
    System.out.println(binarySearch(arr, n, target));
}

public static int binarySearch(int[] arr, int n, int target) {
    int i = 0;
    int j = n - 1;
    while ( i <= j ) {
        int mid = (i + j) / 2;
        if ( arr[mid] == target ) {
            return mid;
        } else if ( arr[mid] < target ) {
            i = mid + 1;
        } else if ( arr[mid] > target ) {
            j = mid - 1;
        }
    }
    return i;
}
```



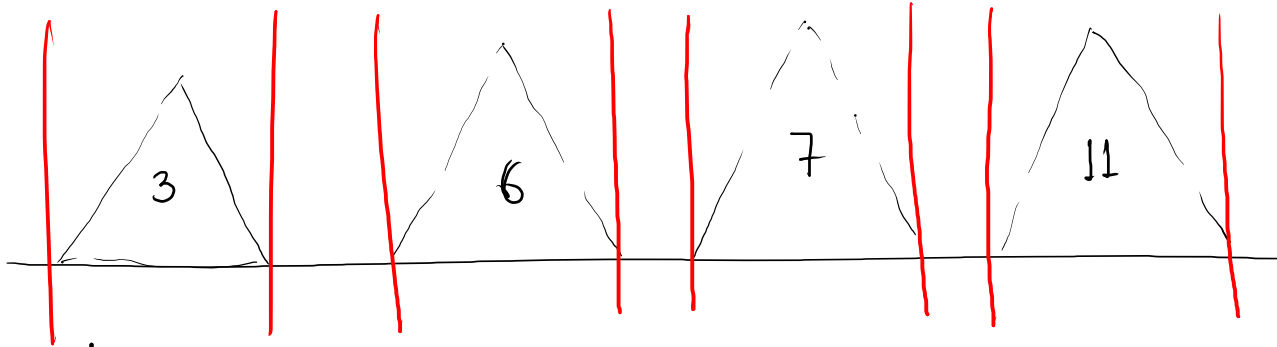
single word changed
from template

The banana challenge

$$\underline{\underline{h = 8}}$$

arr =

0	1	2	3
3	6	7	11



Imp points:-

- ↳ we have 'h' hours only
- ↳ n group of banana's with arr[i] banana's
- ↳ find speed of eating banana's
- ↳ within 1 hour, we can choose only 1 pile.

Imp