

## 20.5 Range

## Range

Given an array of integers **a** sorted in ascending order, find the starting and ending position of a given **n** target value. Your algorithm's runtime complexity must be in the order of  $O(\log n)$ .

```
abstract class RangeBase{
    protected IntUtil u = new IntUtil();
    protected int numSteps = 0 ;
    protected boolean show = true ;
```

abstract int[] Range(int [] a, int n) ;

sorted

n = 2

0	1	2	3	n=2	
1	2	2	3		returns {1,2}
					numSteps = 5

  

0	1	2	3	n=4	
1	2	2	3		returns{-1,-1}
					numSteps = 3

  

0	1	2	3	n=2	
1	2	3	4		returns{1,1}
					numSteps = 4

  

0	1	2	3	n=1	
1	1	1	1		returns{0,3}
					numSteps = 5

```
{
    int N = 1024 ;
    int M = 5 ;
    int [] a = new int[N] ;
    for (int i = 0; i < N; ++i) {
        a[i] = M ;
    }
    returns {0,1023}
    numSteps = 21
```

$\text{numSteps} = O(\log_2 n)$

Figure 20.4: Range

## 20.6 Sum of two numbers equal to N