

s1:

```
// A Java program to find floor(sqrt(x))
```

```
class GFG {
```

```
    // Returns floor of square root of x
    static int floorSqrt(int x)
    {
        // Base cases
        if (x == 0 || x == 1)
            return x;

        // Starting from 1, try all numbers until
        // i*i is greater than or equal to x.
        int i = 1, result = 1;

        while (result <= x) {
            i++;
            result = i * i;
        }
        return i - 1;
    }
}
```

```
    // Driver program
    public static void main(String[] args)
    {
        int x = 11;
        System.out.print(floorSqrt(x));
    }
}
```

Time complexity of s1  $O(n^{0.5})$

s2:

```
// A Java program to find floor(sqrt(x))
```

```
public class Test
```

```
{
```

```
    public static int floorSqrt(int x)
    {
        // Base Cases
        if (x == 0 || x == 1)
            return x;

        // Do Binary Search for floor(sqrt(x))
        int start = 1, end = x, ans=0;
        while (start <= end)
        {
            int mid = (start + end) / 2;
```

```

    // If x is a perfect square
    if (mid*mid == x)
        return mid;

    // Since we need floor, we update answer when mid*mid is
    // smaller than x, and move closer to sqrt(x)
    if (mid*mid < x)
    {
        start = mid + 1;
        ans = mid;
    }
    else // If mid*mid is greater than x
        end = mid - 1;
}
return ans;
}

```

```

// Driver Method
public static void main(String args[])
{
    int x = 11;
    System.out.println(floorSqrt(x));
}
}

```

Time complexity of s2  $O(\log n)$