

补种未成活胡杨

题目描述：

近些年来，我国防沙治沙取得显著成果。某沙漠新种植 N 棵胡杨（编号 $1-N$ ），排成一排。

一个月后，有 M 棵胡杨未能成活。

现可补种胡杨 K 棵，请问如何补种（只能补种，不能新种），可以得到最多的连续胡杨树？

输入描述：

N 总种植数量

M 未成活胡杨数量

M 个空格分隔的数，按编号从小到大排列

K 最多可以补种的数量

其中：

$1 \leq N \leq 100000$

$1 \leq M \leq N$

$0 \leq K \leq M$

输出描述：

最多的连续胡杨棵树

补充说明：

示例 1

输入：

5

2

2 4

1

输出：

3

说明：

补种到 2 或 4 结果一样，最多的连续胡杨棵树都是 3

示例 2

输入：

10

3

2 4 7

1

输出：

6

说明：

补种第 7 棵树，最多的连续胡杨棵树为 6(5,6,7,8,9,10)

```
import java.util.Arrays;
```

```
import java.util.LinkedList;
```

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int n = sc.nextInt();
```

```
        int m = sc.nextInt();
```

```
        int[] arr = new int[n];
```

```
        Arrays.fill(arr, 1);
```

```
        for (int i = 0; i < m; i++) {
```

```
            int index = sc.nextInt();
```

```
            arr[index - 1] = 0;
```

```
        }
```

```
        int k = sc.nextInt();
```

```
        System.out.println(getMaxLen(arr, n, k));
```

```
    }
```

```
    private static int getMaxLen(int[] arr, int n, int k) {
```

```
        int ans = Integer.MIN_VALUE;
```

```

int l = 0;
int r = 0;
LinkedList<Integer> leftIndex = new LinkedList<>();
while (r < n) {
    if (arr[r] == 0) {
        leftIndex.add(r);
        if (leftIndex.size() > k) {
            ans = Math.max(ans, r - l);
            l = leftIndex.removeFirst() + 1;
        }
        r++;
    } else {
        r++;
    }
}
ans = Math.max(ans, r - l);
return ans;
}
}

```