

跳格子 2 题目描述：

小明和朋友玩跳格子游戏，有 n 个连续格子组成的圆圈，每个格子有不同的分数，小朋友可以选择从任意格子起跳，但是不能跳连续的格子，不能回头跳，也不能超过一圈；

给定一个代表每个格子得分的非负整数数组，计算能够得到的最高分数。

输入描述：

给定一个数例，第一个格子和最后一个格子收尾相连，如：2 3 2

输出描述：

输出能够得到的最高分，如：3

补充说明：

```
1 <= nums.length <= 100
```

```
0 <= nums[i] <= 1000
```

示例 1

输入：

2 3 2

输出：

3

说明：

只能跳 3 这个格子，因为第一个格子和第三个格子收尾相连

示例 2

输入：

1 2 3 1

输出：

4

说明：

$1+3=4$

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

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

```
// 注意类名必须为 Main, 不要有任何 package xxx 信息
```

```
public class Main {
```

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

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

```
        String s = scanner.nextLine();
```

```
        String[] split = s.split(" ");
```

```
        int[] ints = Arrays.stream(split).mapToInt(Integer::parseInt).toArray();
```

```
        System.out.println(result(ints));
```

```
    }
```

```
    public static int result(int[] nums) {
```

```
        int length = nums.length;
```

```
        if (length == 1) {
```

```
            return nums[0];
```

```
        } else if (length == 2) {
```

```
            return Math.max(nums[0], nums[1]);
```

```
        }
```

```
        return Math.max(tableRange(nums, 0, length - 2), tableRange(nums, 1, length - 1));
    }

    public static int tableRange(int[] nums, int start, int end) {
        int first = nums[start];
        int second = Math.max(nums[start], nums[start + 1]);
        for (int i = start + 2; i <= end; i++) {
            int temp = second;
            second = Math.max(first + nums[i], second);
            first = temp;
        }
        return second;
    }
}
```