Stack 过时 ： <https://blog.csdn.net/devnn/article/details/82716447>

HashMap的遍历：https://www.cnblogs.com/zmsn/p/12201013.html

Bfs =》找最短路径

Dp：

1. 状态保存的是什么？
2. 如何从这个状态转移到下一个状态
3. 边界

Dfs：

1. 截止条件
2. 遍历候选节点
3. 筛选候选节点

Slide window

<https://blog.csdn.net/lenfranky/article/details/89639320>

各种类型的转换

LC 208 字母（26个）与int的转换, 直接 – ‘a’

<https://www.jianshu.com/p/dd348a1fa51f>

默认（最小堆）,可以加重复元素

PriorityQueue<ListNode> minPQ = **new** PriorityQueue<>(**new** Comparator<ListNode>(){  
 **public int** compare(ListNode l1, ListNode l2){  
 **return** l1.**val** - l2.**val**;  
 }  
});

PriorityQueue<ListNode> maxPQ = **new** PriorityQueue<>(**new** Comparator<ListNode>(){  
 **public int** compare(ListNode l1, ListNode l2){  
 **return** l2.**val** – l1.**val**;  
 }  
});

Lambda使用： Arrays.sort(intervals, (a,b) ->(a[0] - b[0]));

PriorityQueue<Integer> maxPQ = **new** PriorityQueue<>(

(l1, l2) -> l2.compareTo(l1)

);

PriorityQueue<Integer> maxPQ = **new** PriorityQueue<>((l1, l2) -> {  
 **return** l2.compareTo(l1);  
});

**public class** sort {  
 **private static class** s **implements** Comparator<Integer> {  
 **public int** compare(Integer a, Integer b){  
 **return** a - b;  
 }  
 }  
 **public static void** main(String[] args){  
 Integer[] arr = {3,4,2,5,98,100};  
 Arrays.*sort*(arr,**new** s());  
 **for**(**int** i: arr){  
 System.***out***.println(i); 🡺{升序排列，2，3，4，5，98，100}  
 }  
 }  
}

q.offer(new int[]{cury, curx}); //505题，直接新建数组进队列的简便写法

return result.toArray(new int[result.size()][]); Arraylist<int[]> 转换到 int[][]的写法 //986题

佛洛依德环探测（乌龟和兔子）： 287，142

String s 与 String t 比较，请使用 s.equals(t) 而不是 s == t

//int[][]的排序, 406题

Arrays.sort(people,(a, b) ->  
 {  
 *//身高按降序排列* **if**(b[0] != a[0]){  
 **return** b[0] - a[0];  
 }  
 *//身高相同的情况下，按k升序排列* **else**{  
 **return** a[1] - b[1];  
 }  
 }  
);