```
HomeWork_Lecture1_Array+LinkedList+String_部分题目思路图解
3/28/17 8:37 PM
```

```
String 2_38_Count and Say
```

基本思路:以n为例,直接从左向右扫描n-1字符串,计算出现相同数字的个数,直至扫描结束。

```
B示如下:

Start point(i)

Int count=1

first different number, let count=1, let Suy = S. charAt(i)

Count+1

Chart Say = S. charAt(0);

String Builder Sb = new String Builder();

Sb. append (Count);

Sb. append (Say);

Count = 1;

Say = S. charAt(i);

Sb. append (count);
```

Sb. append (say);

(sur! = );

Sery = S. charAe(i);

Sb. append (come);

Sb. uppend (Say).

初始:count=1,每遇到数字相同,count++,然后将count转为字符串或字符类型+当前数组字符以此为规律找出nth字符串。

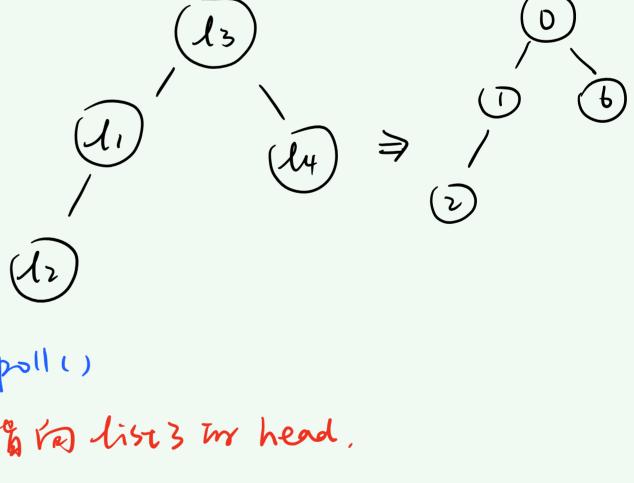
## LinkedList 5\_23\_Merge K Sorted List

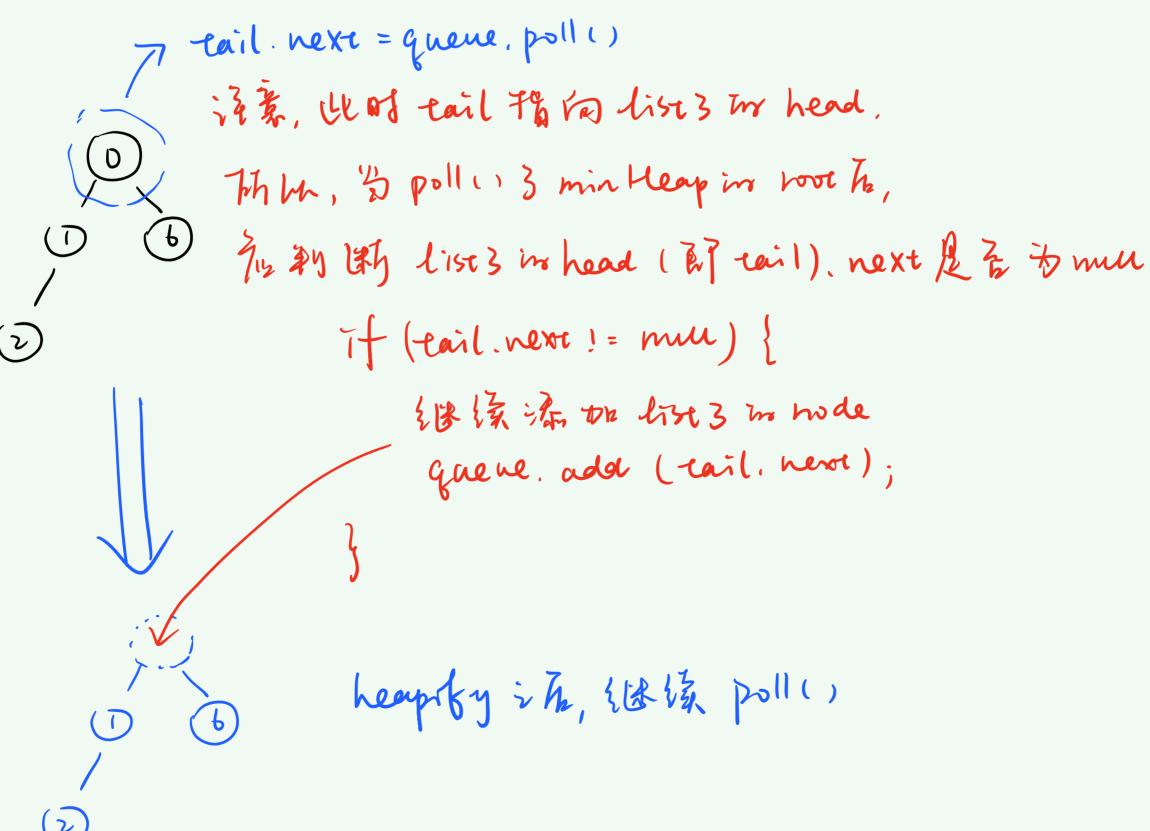
## 图示如下:

```
if (lists==null||lists.size()==0) return null;
    PriorityQueue<ListNode> queue= new PriorityQueue<ListNode>(lists.size(),new Comparator<ListNode>(){
                                                                                                             list3 rext
        @Override
        public int compare(ListNode o1,ListNode o2){
           if (o1.val<o2.val)</pre>
                                                              list 1
               return -1;
           else if (o1.val==o2.val)
               return 0;
                                                              list 2
           else
               return 1;
                                                              list
    });
                                                              list 4
    ListNode dummy = new ListNode(0);
    ListNode tail=dummy;
    for (ListNode node:lists)
        if (node!=null)
           queue.add(node);
    while (!queue.isEmpty()){
        tail.next=queue.poll();
        tail=tail.next;
        if (tail.next!=null)
         quere.add(tail.next);
return dummy.next; [isl ]
```

lise 2 (12): (D) → (4)

lise 3 (13): 0 > 5 17se 4 (14): 1 > 3





## 基本思路:从后往前merge,逆向思维。

Array1\_88\_Merge Two Sorted Array

