

09:05 正式开始上课

上午对应的讲义：《位运算》《海量数据场景下常见算法问题》、《字符串》

下午对应的讲义：《查找》讲义第二部分,《数学与数论》《海量数据场景下常见算法问题》

作业

1.你熟悉的语言是如何使用字符串

2.转换专题

`int = convertStringToInt(string)`

将字符串讲义2.1.2 字符串转换整数 里题目列举的几种情况背下来!!!

3.不用看：字符串匹配

`longStr=aaaabcvbswrffwfweweeer`

`shortStr=few`

KMP

4.理解堆的结构和特征，以及堆如何实现查找和排序的。

哪些算法适合使用堆解决

5.数学部分最重要的两个主题：溢出，进制

上午内容 位运算->海量数据场景下的常见算法

树的重要拓展：

二叉查找：二叉树的中序遍历的特殊情况

排序：快速排序：整体框架是二叉树的前序遍历+数组的对撞型双指针(快慢型，对撞型)

数组中第K大的数字为什么可以用快速排序来解决：

每循环一次对撞调整，哨兵的位置都唯一确定了，此时可以根据哨兵的索引与所有元素的个数来计算哨兵在第几大位置上，

然后就可以根据该值与K的关系来判断下一步递归时处理数组的左侧还是右侧。


```
000000000000000000000000 0000 0110 10 1 (a>>i)&0x01
& 000000000000000000000000 0000 0000 000 1
```

```
000000000000000000000000 0000 0110 10 1
& 000000000000000000000000 0000 0000 001 0
000000000000000000000000 0000 0000 00 0
```

方法1：可以左移0x01,再与原始数据与运算

方法2：可以右移原始数据,再与0x01与运算

2.2 位在海量数据存储中的应用

存储数据的基本思想：

每个位置代表一个数有的话就标记为1

堆的处理

14:05 正式开始上课

3. 字符串

重点内容：

1.熟悉不用语言的具体实现方式、方法

2.重点题目：字符串转成数字需要考虑的几种情况，以及如何实现

在算法面试中，很多时候需要自己将字符串转换成字符数组，进行处理，完成之后，再转换成String

```
char[] arr = s.toCharArray();
返回的时候
return new String(arr);
```

4.数字与数学

第一个模板：整数获得某个位置的元素：N表示进制

```

while(M!=0){
    temp=M%N;
    //技巧一：通过数组F[]解决了大量繁琐的不同进制之间映射的问题
    sb.append(F[temp]);
    M=M/N;
}

```

参考题目：2.3.1 七进制数

第二个模板：溢出的处理模板：

```

if (res>214748364 || (res==214748364 && tmp>7)) {
    return 0;
}
//判断是否小于最小的32位整数
if (res<-214748364 || (res==-214748364 && tmp<-8)) {
    return 0;
}

```

更优雅的方式：

```

// 【4.2】提前判断乘以10以后是否越界,但res*10可能会越界,所以这里使用
Integer.MAX_VALUE/10, 这样一定不会越界。
if (res > Integer.MAX_VALUE / 10 || (res == Integer.MAX_VALUE / 10 &&
(currChar - '0') > Integer.MAX_VALUE % 10)) {
    return Integer.MAX_VALUE;
}
if (res < Integer.MIN_VALUE / 10 || (res == Integer.MIN_VALUE / 10 &&
(currChar - '0') > -(Integer.MIN_VALUE % 10))) {
    return Integer.MIN_VALUE;
}

```

第三个处理ascii字符转换的模板

"01234569"

例如：char '1' 如何转换 1

```

char currChar = charArray[index];
currChar= '1'-'0'
(currChar - '0')

```

三种常见的转换，就可以通过距离来转换成数字

字符转成数字 `currChar= x - '0'`

下面的作用是将字符放到一个初始索引为0的数组里

小写currChar= x - 'a'

大写currChar= x - 'A'

"abcdef"

x在数组中的位置x-'a'

经典使用案例：滑动窗口一章的2.5 字符串的排列

5. 堆的问题

1.堆的特征（大/小，规则一致，条件相反）

2.堆如何插入和删除结点

3.通过堆进行查找（只是更新结点，不再增删结点），维持堆大小不变操作有差异

堆查找适合什么情况：

- 第一种：从超大规模数据中找有限个元素，例如topK，第K大，堆的大小就稳定在K，不会持续增长
- 第二种：流数据，只能用堆

堆查找的特性：

- 可以实时返回从开始位置到当前位置的第K大(就是根元素)，或者topK（就是堆里的全部元素）。
- 时间复杂度 $n \log K$, nk