

670.最大交换

题目大意

给定一个整数，你可以选择交换某两位的数字，求交换后，这个数字最大可能变成多少。

题目分析

实际上，我们需要找到最大位（如果数字相同，取尽量靠右的那一个），然后把它和它左边的某个最小位交换（如果数字相同，取尽量靠左的那一个）。

简略的证明如下：fix交换的其中一个元素，这个元素是两者中较小的，考察另一个元素怎么取更优：如果取了左边的，这个数会变小，因此一定要取右边的。如果数字相同，越往右取，增大的越多。可以用类似的方法证明另一条性质（左边最小位）。

问题在于，有可能出现某一最大元素的左侧没有任何比他小的元素，这时候需要在它的右侧找最大的元素，重复这一操作直至在最大元素的左侧找到比它小的元素，由于输入数据在 $1e8$ 以内，即只有8位，便不再考虑复杂度限制，可以乱写一通...但对于大整数，可以采用支持RMQ的数据结构对找最大最小值的操作进行优化。本题代码如下：

```
int maximumSwap(int num) {
    int temp = num;
    int n = 0;
    while (temp) {
        n++;
        temp /= 10;
    }
    int a[10];
    for (int i = n; i >= 1; i--) {
        a[i] = num % 10;
        num /= 10;
    }
    int ptr = 0;
    bool flag = false;
    while (!flag and ptr <= n) {
        ptr++;
        int maxpos = -1, maxnum = -1;
        for (int i = ptr; i <= n; i++) {
            if (a[i] >= maxnum) {
                maxpos = i;
                maxnum = a[i];
            }
        }
    }
```

```
    }  
    for (int i = 1; i <= maxpos; i++) {  
        if (a[i] < maxnum) {  
            swap(a[i], a[maxpos]);  
            flag = true;  
            break;  
        }  
    }  
}  
  
int ret = 0;  
for (int i = 1; i <= n; i++) ret = (ret * 10) + a[i];  
return ret;  
}
```