

给定一个**32位**正整数 n ，你需要找到最小的**32位**整数，其与 n 中存在的位数完全相同，并且其值大于 n 。如果不存在这样的**32位**整数，则返回-1。

示例 1:

输入: 12

输出: 21

示例 2:

输入: 21

输出: -1

思路: `int`转换为字符串，从后往前遍历字符串，找到最大的数，若是当前的数比最大的数小，那么就找刚好比当前数大的数，交换到当前数的位置上，然后后面按照升序排列即可。需注意`int`倒转时，数据超过`int`的最大限制。

代码:

```
class Solution {
public:
    int nextGreaterElement(int n) {
        if(n<10){
            return -1;
        }
        string num=to_string(n);
        string maxnum="2147483647";
        map<char,int> bj;
        char max='0'-1;
        for(int i=num.size()-1;i>=0;i--){
            bj[num[i]]++;
            if(num[i]>=max){
                max=num[i];
            }else{
                char j;
                for(j=num[i]+1;j<='9';j++){
                    if(bj[j]>0){
                        bj[j]--;
                        num[i]=j;
                        break;
                    }
                }
            }
        }
        int m=num.size()-1;
        for(char k='9';k>='0';k--){
            while(bj[k]>0){
                num[m]=k;
                bj[k]--;
            }
        }
    }
};
```

```
        m--;
    }
}
if(num.size()==maxnum.size()){
    if(num>maxnum){
        return -1;
    }
}
return atoi(num.c_str());
}
return -1;
}
};
```