第三次作业说明文档

> 1950698 陆天宇

第一题

题目描述

> Given an integer , return the count of all numbers with unique digits , where $0 \leq n < 10$ and $0 \leq x < 10^n$. Please give a Dynamic Programming solution.

解题思路

1. 先考虑0这个特殊情况
2. 然后先考虑1位的情况，1位有9种可能（不能是0），2位就是把1位的末尾填上一个数，前面已经确定了一个情况那补的那个数就只有10-1=9种情况。
3. 以此类推，n位的所有情况就是n-1位的情况的末尾补上一个数，前面已经确定了n-1个数，那么就只有10-n+1=11-n个情况。由此可得状态转移方程。然后把1位的情况累加到n位的情况就是答案。

$$dp[0] = 0 \quad dp[1] = 9$$
$$dp[n] = dp[n-1] * (11-n)$$
$$answer = \sum_{i=0}^{n} dp[i]$$

时间复杂度:

代码实现

因为状态只在i和i-1之间转移，为了节省空间没有必要开大的数组。用一个int变量表示就行了。

```
if (n==0) return 1;
else if (n==1) return 10;

int sum = 10;
int dp = 9;
for (int i = 2; i <= n; i++)
{
    dp *= 11 - i;
    sum += dp;
}
cout<<sum;
```

力扣OJ结果

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	0 ms	5.8 MB	C++	2021/06/01 08:45	添加备注

第二题

题目描述

> Given an array of intervals intervals where $intervals[i] = [start_i, end_i]$, return the minimum number of intervals you need to remove to make the rest of the intervals non-overlapping.

解题思路

贪，都可以贪。

1. 按照end排序
2. 按照end从小到大访问，只要能放进去，就放进去，因为保证了end的递增，只要前面的end小，后面能放进去的概率越大（显然）。
3. 然后总数减去放进去的数就是删掉的数。

时间复杂度： $O(nlogn)$

代码实现

```
while(cin>>a>>b){
    elm.push_back(pair<int,int>(a,b));
}
sort(elm.begin(),elm.end());
int end=elm[0].second;
int ans=1;
for(int i=1;i<elm.size();i++)
{
    if(elm[i].first>=end)
```

```
    {
        end=elm[i].second;
        ans++;
    }
}
cout<<elm.size()-ans;
```

力扣OJ结果

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	16 ms	10 MB	C++	2021/06/01 09:12	添加备注

解题思路(dp)

1. 排序，排完序之后分别为$[l_0,r_0][l_1,r_1][l_2,r_2]...[l_n,r_n]$
2. 令f_i表示以区间i作为最后一个区间能选出来的区间数量的最大值。
3. 状态转移方程

$$f_i = max_{j<i\&\&r_j\leq l_i}\{f_j\}+1$$

时间复杂度：$O(n^2)$

代码实现(dp)

```
sort(elm.begin(),elm.end());
int n=elm.size();
vector<int> dp(n,1);
for(int i=0;i<n;i++)
{
    for(int j=0;j<i;j++)
    {
        if (elm[j].second <= elm[i].first) {
            dp[i] = max(dp[i], dp[j] + 1);
        }
    }
}
cout<<n-*max_element(dp.begin(),dp.end());
```

力扣OJ结果（dp）

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	1176 ms	9.9 MB	C++	2021/06/01 09:56	添加备注

可见有时候该贪还是得贪。