**绍兴一中练习赛（1.21）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **题目名称** | **提交文件** | **输入文件** | **输出文件** | **时间限制** | **空间限制** |
| **约会** | **dating.\*** | **dating.in** | **dating.out** | **1s** | **128M** |
| **跳跃数** | **jnum.\*** | **jnum.in** | **jnum.out** | **1s** | **128M** |
| **括号匹配** | **match.\*** | **match.in** | **match.out** | **1s** | **128M** |
| **停车问题** | **park.\*** | **park.in** | **park.out** | **1s** | **128M** |

**这次考试光荣爆蛋。。。。。。**

**第一题**

1. **看一下数据范围，一看就是模拟。**

**考试时我很傻逼地搞了一个链表，打到一半发现要用双向链表，然后又打了一个寻找前链的函数，等样例过后，代码100行+，时间60分+，然后爆蛋；**

**其实用一个bool数组标记一下这个点有没有出队就可以了，毕竟n只有50；**

**但是体面没说k的范围，太坑了，当k十分大时，需要取%；**

**比如k=100000;**

**一次找一对，100000.太慢了；**

**其实n不大，当k大于n时，我们只要模拟k%n就可以了，如果不%，k其实在不断转圈；**

**当然每次一对人出队，n会-2**

**所以每次k%n时，n会不断减小；**

**还有当n=0时，不可取；**

**还有，k%n==0时，k其实是n；**

**第二题**

**这钟类型的题目初赛时我是过的，用组合公式，但是我……不会**

**后来才知道，是dp啊**

**出入：l r**

**我们要求出l~r间的跳跃数；**

**直接求是很难滴，我们搞一个函数get（x）返回0~x的跳跃数；**

**答案就是get(r)-get(l-1);**

**当然咯直接取get也是很难滴**

**我们搞一个f[i][j]数组表示第i位上是j的（位数只能小于等于i位）的所有跳跃数。这里j是可以是0的，因为我第i位是0，只要i+1位非0，这个仍然成立；**

**for(int i=0;i<=9;i++)f[1][i]=1;**

**for(int i=2;i<=10;i++)**

**for(int j=0;j<=9;j++)**

**for(int k=0;k<=9;k++)**

**if(abs(k-j)>1)f[i][j]+=f[i-1][k];**

**这个abs就在满足跳跃数的条件**

**Get函数就是把不同的f[][]加起来返回一个值**

**long long get(long long x)**

**{**

**首先，如果x小于10，返回x+1就好了（因为有个0，所以+1）;**

**然后贼麻烦**

**比如556**

**我们先取0~99**

**再取100~499**

**最后取**

**500~556**

**每位（除个位）割去a[i]-1;**

**500-540**

**541-549**

**550-556**

**预处理 n为x的位数**

**a[i]是x第i位的数**

**for(int i=1;i<=n-1;i++)**

**for(int j=1;j<=9;j++)ans+=f[i][j];**

**为什么j是1~9呢，因为这是在当位数只有i位时，首位不可以为0**

**但是光一个0也是是跳跃数，所以返回值ans一开始要为1**

**这样可以求出位数在1~n-1的个数6666666**

**因为当数的位数小于n时取数不受限制**

**受限制就是说取到的数不可以大于x**

**for(int i=1;i<=a[n]-1;i++)ans+=f[n][i];**

**求出位数是n，但首位不到a[n]的数，也不受限制**

**6666666666666666666666666666666666666666666666666666**

**for(int i=n-1;i>=1;i--){**

**for(int j=0;j<=a[i]-(i!=1);j++)**

**if (abs(j-a[i+1])>=2)ans+=f[i][j];**

**if (abs(a[i]-a[i+1])<2) return ans;**

**}**

**这个东西就是求**

**500-540**

**541-5496**

**550-556**

**i是位数，j是第i位的数字**

**我们要知道 当i不断减小，i-1位上的数字一定是a[i-1]**

**比如13131**

**我们搞定1去取1\*\*\*\*的所有有可能**

**再高定13去取13\*\*\*的所有可能**

**再高定13去取13\*\*\*的所有可能**

**再高定131去取131\*\*的所有可能**

**再高定1313去取1313\*的所有可能**

**13131包括在最后一步**

**但比如112，这个x本身不符合**

**所以循环到1\*\*就直接退出**

**for(int j=0;j<=a[i]-(i!=1);j++)**

**(i!=1)就是判断“13131包括在最后一步”；**

**return ans;**

**}**

**666666666666666666666666666666666666666666666666666666**

**当然也有zz大神临阵打标AC了……………..**

**第三题**

**区间dp啊，我的程序居然爆0，唉~~~**

**对于f[i][j]有两种方法**

1. **f[i+1][i-1]+check(i,j);**

有合法串*S*，则’(*S*)’、’[*S*]’和’{*S*}’也是合法的

1. **for (int k=i+1;k<=j-2;k++)**

**f[i][j]=min(f[i][j],f[i][k]+f[k+1][j]);**

有合法串*S*1和*S*2，则串*S*1*S*2也是合法的。

**很常见的区间dp（不要问我为什么0分）**

**for(int i=n-1;i>=1;i--)**

**for (int j=i+1;j<=n;j++)**

**这个是循环；**

**原先我一直先枚举区间长度，再枚举i，推出j但效率不如这个**

**好了我们来讲一下check函数**

**括号匹配有3种返回值**

**0 1 2**

**[] () {}---0**

**[[ （｝---1**

**][ }{ }(---2**

**我们搞一<map> F**

**F['[']=1;**

**F['(']=2;**

**F['{']=3;**

**F[']']=6;**

**F[')']=5;**

**F['}']=4;**

**int check(int x,int y){**

**if (a[x]<=3&&a[y]>=4&&a[x]+a[y]==7) return 0;**

**if (a[x]>3&a[y]<4) return 2;**

**return 1;**

**}**

**第四题**

**做题目时一脸懵逼**

**后来经fb大神指导原来是贪心水题**

**唉~~~水题都不会做了~~~~**

**每次枚举可以动的点，取最小的移动，再枚举**

**如果全动好了，直接输出答案**

**如果不能动了，NONE**

**首先我们要知道a移动到A时，a的位子其他车是可以过的**

**但A的位置停了a，所以不能动；**

**为什么那个贪心成立呢？？？？**

**666666666666666666666666666666**

**因为给出一个环，所以无论车子怎么移动，车的前后顺序是不会变的**

**a可以到A，b可以到B**

**如果a到A时b不能到B**

**那么如果先移动b到B**

**a也一定无法到A**

**妥妥的**

**-------------------------------------------2017-1-22-----------**

**----------------------------------------------------cfb-------**