动态规划 1

数位DP

算法简介

一般用于处理形如区间 $[l,r]$ 中满足条件的数有几个之类问题。

算法思想

首先考虑将区间 $[l,r]$ 拆分成 $[0,r]-[0,l]$

考虑记忆化搜索，一般搜索函数为 $f(pos,s,eq,zero)$

其中 pos 表示当前搜索位置s 表示前缀状态eq 表示前缀是否与查询上界相等$zero$ 表示是否有前导零。

算法练习

习题一

洛谷p2657

题意

询问区间 $[l,r]$ 中满足相邻数位之差至少为 2 的数的个数。

题解

```
int a[10],dp[10][10];
int dfs(int pos,int pre,bool eq,bool zero){
    if(!pos)return 1;
    if(!eq&&~dp[pos][pre])return dp[pos][pre];
    int ans=0,v=eq?a[pos]:9;
    _rep(i,0,v){
        if(!zero&&abs(i-pre)<2)
        continue;
        ans+=dfs(pos-1,i,eq&&i==a[pos],zero&&i==0);
    }
    if(!eq&&!zero)dp[pos][pre]=ans;
    return ans;
}
int solve(int n){
```

```
    int len=0;
    mem(dp,-1);
    while(n){
        a[++len]=n%10;
        n/=10;
    }
    return dfs(len,0,true,true);
}
int main()
{
    int a=read_int(),b=read_int();
    printf("%d",solve(b)-solve(a-1));
    return 0;
}
```

习题二

洛谷p2602

题意

询问区间 $[l,r]$ 所有整数中 $0\sim 9$ 分别出现的次数。

题解

依次计算 $0\sim 9$ 出现次数。每次把搜索函数中的 s 当成当前前缀的贡献即可。

```
const int MAXL=15;
LL a[MAXL],dp[MAXL][MAXL];
int goal;
LL dfs(int pos,int pre,bool eq,bool zero){
    if(!pos)return pre;
    if(!eq&&~dp[pos][pre])return dp[pos][pre];
    int v=eq?a[pos]:9;LL ans=0;
    _rep(i,0,v){
        if(zero&&i==0)
        ans+=dfs(pos-1,pre,eq&&i==a[pos],true);
        else
        ans+=dfs(pos-1,pre+(i==goal),eq&&i==a[pos],false);
    }
    if(!eq&&!zero)dp[pos][pre]=ans;
    return ans;
}
LL solve(LL n){
    int len=0;
```

```
    mem(dp,-1);
    while(n){
        a[++len]=n%10;
        n/=10;
    }
    return dfs(len,0,true,true);
}
int main()
{
    LL a=read_LL(),b=read_LL();
    _rep(i,0,9){
        goal=i;
        printf("%lld ",solve(b)-solve(a-1));
    }
    return 0;
}
```