Educational Codeforces Round 104

比赛链接

F. Ones

题意

给定 $1\le n\lt 10^{50}$□要求用若干个形如 $11\cdots 11$ 的数以及加减号表示 n□求表达式中字符 1 的个数的最小值。

题解 1

设 $\text{dp}(p,i,j,k)$ 表示最低 p 位与 n 相同，第 $p+1$ 位的进位为 i (允许是负数)。

表达式中有 j 个长度不小于 p 的 $11\cdots 11$ 和 k 个长度不小于 p 的 $-11\cdots 11$ 的方案中仅统计所有数最低 i 位的 1 的个数的最小值。

记 n 的第 p 位为 d□当 $i+j-k\equiv d\bmod 10$ 时，有

$$ \text{dp}(p,\lfloor\frac {i+j-k}{10}\rfloor,j,k)\gets \min_{x\ge j,y\ge k}\text{dp}(p-1,i,x,y)+j+k $$

考虑二维前缀极值维护 $\min_{x\ge j,y\ge k}\text{dp}(p-1,i,x,y)$□于是可以 $O(1)$ 完成递推。

现在考虑状态数，首先 $j,k\le 5\ast\text{len}(n)$□因为可以用不超过 5 个 $11\cdots 11$ 和不超过 5 个 $-11\cdots 11$ 表示出 n 的任意一位。

另外设 i 上界为 T□于是 $|\frac {i+j-k}{10}|\le max\left(\left(|\frac {i+j}{10}|,|\frac {i-k}{10}|\right)\right)\le \frac {T+5\text{len}(n)}{10}\le T$□故 $T\le \frac {5\text{len}(n)}9$□

于是总状态数为 $O(\text{len}(n)^4)$□滚动数组滚掉一维，时间复杂度 $O(\text{len}(n)^4)$□空间复杂度 $O(\text{len}(n)^3)$□

注意需要给 n 补上一个前导 0，否则最高位无法借位，会遗漏形如 $9=11-1-1$ 的情况。

```
const int MAXN=55,MAXV=30,MAXC=255,Inf=1e9;
int a[MAXN],dp[2][MAXV<<1|1][MAXC][MAXC];
char s[MAXN];
int main()
{
    scanf("%s",s);
    int n=strlen(s),pos=0;
    _for(i,0,n)a[i]=s[n-i-1]-'0';
    a[n++]=0;
    _for(i,0,MAXV<<1|1)_for(j,0,MAXC)_for(k,0,MAXC)dp[pos][i][j][k]=Inf;
    dp[pos][MAXV][MAXC-1][MAXC-1]=0;
```

```
    _for(p,0,n){
        pos=!pos;
_for(i,0,MAXV<<1|1)_for(j,0,MAXC)_for(k,0,MAXC)dp[pos][i][j][k]=Inf;
        _for(i,0,MAXV<<1|1)for(int j=MAXC-1;j>=0;j--)for(int
k=MAXC-1;k>=0;k--){
            if(dp[!pos][i][j][k]==Inf)continue;
if(j)dp[!pos][i][j-1][k]=min(dp[!pos][i][j-1][k],dp[!pos][i][j][k]);
if(k)dp[!pos][i][j][k-1]=min(dp[!pos][i][j][k-1],dp[!pos][i][j][k]);
            int i2=i-MAXV;
            if((i2+j-k-a[p])%10==0){
                int next_i=(i2+j-k-a[p])/10+MAXV;
dp[pos][next_i][j][k]=min(dp[pos][next_i][j][k],dp[!pos][i][j][k]+j+k);
            }
        }
    }
    int ans=Inf;
    _for(i,0,MAXC)_for(j,0,MAXC)ans=min(ans,dp[pos][MAXV][i][j]);
    enter(ans);
    return 0;
}
```

题解 2

玄学做法，设 $m_i= \underbrace{11\cdots 1}_i$，从高到低考虑使用每个 m_i，每次操作可以认为是 $n\to n+m_i$ 或 $n\to n-m_i$。

其中 $\text{dp}(pos,v,c,f)$ 表示考虑到 m_{pos}，n 的高位剩下 $v\times 10^{pos}$，

c 表示所有长度不小于 pos 的 m_i 对第 pos 位的贡献，f 表示当前操作是 $+m_i$ 还是 $-m_i$。

于是 $\text{dp}(pos,v,c,f)\gets \text{dp}(pos,v,c+f,f)$ 表示再选一个 m_{pos}，记 n 的第 pos 位最开始为 d。

于是 $\text{dp}(pos,v,c,f)\gets\min(\text{dp}(pos-1,10*v+c-d,c,1),\text{dp}(pos+1,10*v+c-d,c,1))$ 表示考虑 m_{pos+1}。

关于 c 的上界，由题解一论证知 $|c|\le 5\ast \text{len}(n)$。

关于 v，猜测需要用 m_i 将 n 转化为绝对值小于 m_i 的数再考虑 m_{i+1}。

于是 n 的第 pos 位最后仅允许是 $0,\pm 1$，而第 pos 位的实际值为 $10*v+c-d+\lfloor\frac c{10}\rfloor$。

新 v 为 $10*v+c-d$，于是新 v 不超过 $\lfloor\frac c{10}\rfloor\pm 1\sim \text{len}(n)$，总时间复杂度 $O(\text{len}(n)^3)$。

```
const int MAXN=55,MAXV=50,MAXC=255,Inf=1e9;
int n,a[MAXN],dp[MAXN][MAXV<<1|1][MAXC<<1|1][2];
```

```
char s[MAXN];
int dfs(int pos,int v,int c,int f){
    if(!pos)return v==0?0:Inf;
    if(v<-MAXV||v>MAXV||c<-MAXC||c>MAXC)return Inf;
    if(~dp[pos][v+MAXV][c+MAXC][f==1])return dp[pos][v+MAXV][c+MAXC][f==1];
    int d=min(dfs(pos-1,v*10+c-a[pos-1],c,1),dfs(pos-1,v*10+c-
a[pos-1],c,-1));
    return dp[pos][v+MAXV][c+MAXC][f==1]=min(dfs(pos,v,c+f,f)+pos,d);
}
int main()
{
    scanf("%s",s);
    n=strlen(s);
    _for(i,0,n)a[i]=s[n-i-1]-'0';
    a[n++]=0;
    mem(dp,-1);
    enter(dfs(n,0,0,1));
    return 0;
}
```

From:
https://wiki.cvbbacm.com/ - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:edu_104&rev=1613566230

Last update: **2021/02/17 20:50**