说明文档

问题描述

Problem: Locker

A password locker with N ($N \leq 1000$) digits, each digit can be rotated to 0-9 circularly. You can rotate 1-3 consecutive digits up or down in one step. For examples:

```
567890 → 567901 (by rotating the last 3 digits up)
000000 → 000900 (by rotating the 4th digit down)
```

Given the current state and the secret password, what is **the minimum amount of steps** you have to rotate the locker in order to get from current state to the secret password?

一个锁，有密码，每次只能转连续的1到3位，问从一个状态转到另一个状态最少的步数。

解题思路

看这个N只有1000，每次转动只涉及到三位的变化，而且每次只变0-9，那么状态就可以表示为status(n,a,b,c),表示当前在第n位，当前位置为a，下一位为b，下一位为c，不管是转一个，转两个，还是转三个，都是a转到位之后，b和c在a转动的过程中转移到status(n+1,b+i,c+j,d)，(0<=j<=i<=a转动圈数）或者status(n+1,(b-i+10)%10,(c-j+10)%10,d)，(0<=j<=i<=a转动圈数）。然后用一个数组记录一下每个状态的结果，记忆化搜索。

- 时间复杂度：$O(1000n)$
- 空间复杂度：$O(1000n)$

代码实现

记忆化搜索部分

```
    /*********************************************
     * 记忆化搜索
     * @param cur 当前位置
     * @param x 当前位置的值
     * @param y 下一位的值
     * @param z 再下一位的值
     * @return 当前状态最少步数
     *********************************************/
    int dp(int cur, int x, int y, int z)
    {
        if (cur >= n) return 0;
        int &res = f[cur][x][y][z];
        if (vis[cur][x][y][z]) return res;
        vis[cur][x][y][z] = true;
        res = inf;
        int delta;
        if (x <= b[cur]) delta = b[cur] - x;
        else delta = b[cur] + 10 - x;
```

```
        for (int i = 0; i <= delta; i++) {
            for (int j = 0; j <= i; j++) {
                res=min(res, dp(cur + 1, (y + i) % 10, (z + j) % 10, a[cur + 3])
+ delta);
            }
        }
        if (x >= b[cur]) delta = x - b[cur];
        else delta = x + 10 - b[cur];
        for (int i = 0; i <= delta; i++) {
            for (int j = 0; j <= i; j++) {
                res=min(res, dp(cur + 1, (y - i + 10) % 10, (z - j + 10) % 10,
a[cur + 3]) + delta);
            }
        }
        return res;
    }
```

主体函数

```
    void solve()
    {
        scanf("%s %s", s1, s2);
        //cout<<s1<<"\t"<<s2<<endl;
        n = strlen(s1);
        for (int i = 0; i < n; i ++) a[i] = s1[i] - '0', b[i] = s2[i] - '0';
        a[n] = a[n + 1] = b[n] = b[n + 1] = 0;//减少结束条件的判断。
        memset(vis,0,sizeof(vis));
        printf("%d\n", dp(0, a[0], a[1], a[2]));
    }
```

运行结果

```
1234567890 0987654321
16
```

```
15102342 15123094
8
```