题意：给你一个数，你可以交换这个数的任意两个相邻的数字，但不能出现前导零，问能得到被25整除的数的最少交换次数是多少

当时做的时候毫无思路

其实就是朴素暴力，一共才18位嘛，枚举嘛

用i和j来枚举移动到最后一位的数字和移动到倒数第二位的数字

如果出现前导零，就把最左边不是零的数字移动到第一位

枚举找最小交换次数

注意，j也要从0开始枚举而不能从i+1开始，因为i和j枚举的数字放的位数是不同的，i是放到最后一位而j是放在倒二位

还有，如果j下标是在i之后的话，i移动到后面之后，j代表的那个数自动往前移了一位，这时下标j要减一

#include <iostream>

#include <cstdio>

#include <cstring>

#include <algorithm>

using namespace std;

typedef long long ll;

#define INF 1000000007

int main()

{

// freopen("input.txt","r",stdin);

string s;

cin>>s;

int l=s.size();

int ans(INF);

for(int i=0;i<l;i++)

for(int j=0;j<l;j++)

{

if(j==i)continue;

int cur=0;

string t=s;

for(int k=i;k<l-1;k++)

{

swap(t[k],t[k+1]);

cur++;

}

int k;

if(j>i)k=j-1;else k=j;

for(;k<l-2;k++)

{

swap(t[k],t[k+1]);

cur++;

}

int p=0;

while(t[p]=='0')p++;

for(int k=p;k>0;k--)

{

swap(t[k],t[k-1]);

cur++;

}

ll n=atoll(t.c\_str());

if(n%25==0)

if(cur<ans)

ans=cur;

}

if(ans==INF)cout<<"-1\n";

else

cout<<ans<<"\n";

return 0;

}

其实还可以更优化

25的倍数的后两位只可能00,25,50,75

#include <iostream>

#include <cstdio>

#include <cstring>

#include <algorithm>

using namespace std;

typedef long long ll;

#define INF 1000000007

string s;

bool pd(int x,int y)

{

if(s[x]=='0' && s[y]=='0')

return 1;

if(s[x]=='2' && s[y]=='5')

return 1;

if(s[x]=='5' && s[y]=='0')

return 1;

if(s[x]=='7' && s[y]=='5')

return 1;

return 0;

}

int main()

{

// freopen("input.txt","r",stdin);

cin>>s;

int l=s.size();

int ans(INF);

for(int i=0;i<l;i++)

for(int j=0;j<l;j++)

{

if(j==i)continue;

if(!pd(j,i))continue;

int cur=0;

string t=s;

for(int k=i;k<l-1;k++)

{

swap(t[k],t[k+1]);

cur++;

}

int k;

if(j>i)k=j-1;else k=j;

for(;k<l-2;k++)

{

swap(t[k],t[k+1]);

cur++;

}

int p=0;

while(t[p]=='0')p++;

for(int k=p;k>0;k--)

{

swap(t[k],t[k-1]);

cur++;

}

ll n=atoll(t.c\_str());

if(n%25==0)

if(cur<ans)

ans=cur;

}

if(ans==INF)cout<<"-1\n";

else

cout<<ans<<"\n";

return 0;

}