【算法分析】

交换k次，其实意味着k个j变成z，k个z变成j。首先相同字符是不用调换的，一个字符最多被调换一次(a<—>b,b<—>c等价于a<—>c)

f[i][j][k]表示前i个字符，改变了j个'j'和k个'z'后的“jz”串数。

只有j==k时，答案才有意义。j!=k的情况是中转状态：可以看做是把前面的某几个字符提了出来，等待之后的交换。

那么只考虑原串中前两位s[i]和s[i-1]，只有四种可能：“jj”“zz”“jz”“zj”，枚举四种情况:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | #include <bits/stdc++.h>  using namespace std;  const int inf =0x7fffffff;  const int mxn=521;  int f[mxn][110][110];  char s[mxn];  int n,K;  int main()  {  memset(f,-0x3f,sizeof f);//注意取值大小  int i,j;  cin>>n>>K;  scanf("%s",s+1);  f[0][0][0]=f[1][0][0]=0;  if(s[1]=='z')  f[1][0][1]=0;  else  f[1][1][0]=0;  for(i=2; i<=n; i++)  {  for(j=0; j<=K; j++)  for(int k=0; k<=K; k++)  {  f[i][j][k]=f[i-1][j][k];  if(s[i-1]=='j' && s[i]=='j' && j)  f[i][j][k]=max(f[i][j][k],f[i-2][j-1][k]+1);  if(s[i-1]=='j' && s[i]=='z')  f[i][j][k]=max(f[i][j][k],f[i-2][j][k]+1);  if(s[i-1]=='z' && s[i]=='z' && k)  f[i][j][k]=max(f[i][j][k],f[i-2][j][k-1]+1);  if(s[i-1]=='z' && s[i]=='j' && j && k)  f[i][j][k]=max(f[i][j][k],f[i-2][j-1][k-1]+1);  }  }  for(i=0; i<K; i++)  f[n][K][K]=max(f[n][K][K],f[n][i][i]);  cout<<f[n][K][K]<<endl;  return 0;  } |