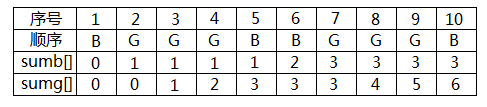
【算法分析】

线性动态规划，可以采取计算前缀和的方式统计同学数，即设sumb[]为男孩总数，sumg[]为女孩总数，例如当同学数为10时，如表6.1所示。

表6.1



　　这样当计算[i，j]段同学是否同一性别时，算式如下：

　int sumboy＝sumb[i]－sumb[j];

int sumgirl＝sumg[i]－sumg[j];

当sumboy或sumgirl为0时，则 [i，j]段同学为同一性别。

判断男生和女生的人数差是否超过k时，算式如下：

abs(sumboy－sumgirl)≤k

设dp[i]表示前i个人使用的最小车厢数，则状态转移方程为：

其中　0≤j＜i，IntMax取最大值。

参考代码如下所示。

|  |  |
| --- | --- |
| 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  41  42  43  44  45  46  47  48  49 | //安排车厢  #include <bits/stdc++.h>  using namespace std;  #define MAXN 1010  #define IntMax 0x3f3f3f3f  int n,m,k;  int sumb[MAXN],sumg[MAXN];  int dp[MAXN];  inline int OK(int j,int i)  {  int sumboy = sumb[i] - sumb[j];  int sumgirl = sumg[i] - sumg[j];  return (i-j<=m) && (sumboy==0 || sumgirl==0 || abs(sumboy-sumgirl)<=k);  }  int main()  {  scanf("%d%d%d",&n,&m,&k);  char stu;  for(int i = 1; i <= n; i++) //初始化并求前缀和  {  cin>>stu; //读字符要小心  if(stu == 'B')  {  sumb[i]=sumb[i-1]+1;  sumg[i]=sumg[i-1];  }  if(stu == 'G')  {  sumg[i]=sumg[i-1]+1;  sumb[i]=sumb[i-1];  }  }  for(int i = 1; i <= n; i++) //动规  {  dp[i]=IntMax;  for(int j =i-1; j>=0; j--)  {  if(!OK(j,i))  break;  dp[i]=min(dp[i],dp[j]+1);  }  }  printf("%d\n",dp[n]);  return 0;  } |