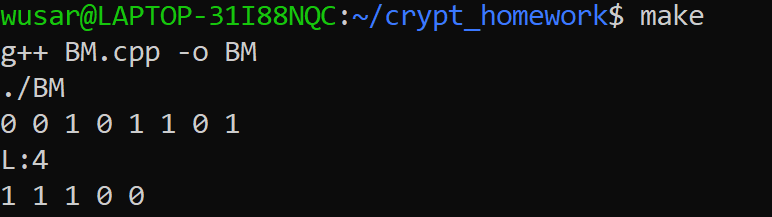


先上实现效果：



可以看出，序列00101101的线性复杂度是4，联结多项式是1+D+D^2.

本程序构造了一个LFSR类：

class LFSR

{

public:

int L;

int C[MAX\_LENGTH], S[MAX\_LENGTH];

public:

LFSR(/\* args \*/);

~LFSR();

void init\_LFSR(int L, int C[], int S[]);

void print();

int LFSR\_next();

int distance(int input);

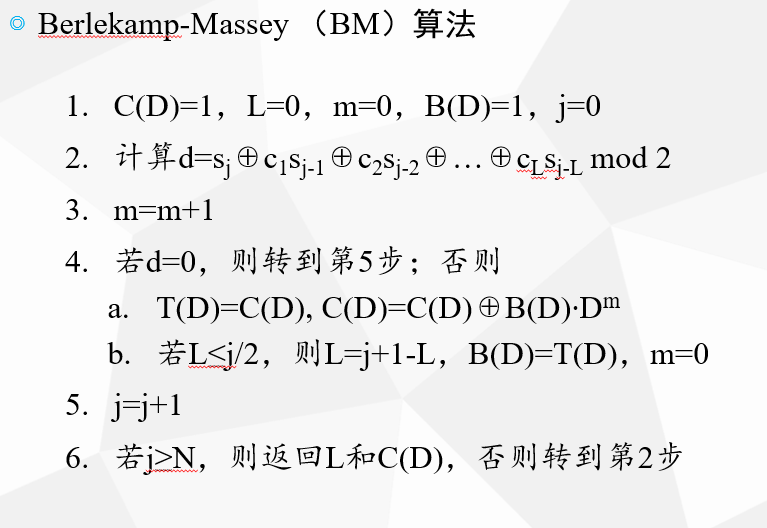
void BM(int sequence[],int length);

};

成员函数void BM(int sequence[],int length)传入一个sequence序列与其长度length，从而使用BM算法算出移位寄存器LFSR的L、C、S

print（）函数打印LFSR的基本信息，包括C、L

BM算法的具体步骤：



这些具体步骤都在BM函数里面得到了实现：

void LFSR::BM(int sequence[],int length)

{

int B[MAX\_LENGTH],T[MAX\_LENGTH];

memset(B,0,sizeof(B));

memset(T,0,sizeof(T));

memset(C,0,sizeof(C));

memset(S,0,sizeof(S));

L=0;

C[0]=1;

B[0]=1;

int m=0;

int j=0;

int d=0;

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

{

d=sequence[i];

for(int j=1;j<L+1;j++)

{

d^=sequence[i-j]\*C[j];

}//求出d

m++;

if(d==0);

else{//若d！=0，对移位寄存器进行更新

for (int j = 0; j < MAX\_LENGTH; j++)

{

T[j]=C[j];

}

for (int j = 0; j < MAX\_LENGTH; j++)

{

if(j-m<0);

else C[j]^=B[j-m];

}

if(2\*L<=i){

L=i+1-L;

for (int j = 0; j < MAX\_LENGTH; j++)

{

B[j]=T[j];

}

m=0;

}

}

}

}