云南大学数学与与统计学院

上机实践报告

课程名称：信息论基础实验	**年级：**2013	**上机实践成绩：**
指导教师：陆正福	**姓名：**金洋	
上机实践名称：线性码实验	**学号：**20131910023	**上机实践日期：**2016/5/6
上机实践编号：No. 7	**组号：**	**上机实践时间：** **12:36**

一、实验目的

（1） 熟悉线性码的基本原理

（2） 通过具体的 Hamming 码的实现以及纠错能力的考察，熟悉纠错码的概念、术语、设计思路和局限性

（3） 熟悉位运算对于二元线性码的实现程序的优化作用

二、实验内容

(1) (15,11,3)Hamming 码编码和译码算法的实现
(2) 通过设置不同的错误模式，考察纠错与检错能力
(3) 考察各列线性无关和线性相关的情况，得出有关最小重量和最小距离的结论。
(4) 程序设计的推广：对于一般的线性码，设计相应的编码程序和译码程序。

三、实验环境

1. 个人计算机，任意可以完成实验的平台，如 Java 平台、Python 语言、R 语言、Matlab 平台、Magma 平台等。
2. 对于信息与计算科学专业的学生，建议选择 Java、Python、R 等平台。
3. 对于非信息与计算科学专业的学生，建议选择 Matlab、Magma 等平台。

四、实验记录与实验结果分析

（注意记录实验中遇到的问题。实验报告的评分依据之一是实验记录的细致程度、实验过程的真实性、实验结果的解释和分析。**如果涉及实验结果截屏，应选择白底黑字。**）

(1) (15,11,3)Hamming 码编码和译码算法的实现

Hamming.java

```
package IT7;

import java.util.ArrayList;

public class Hamming {


    /*校验矩阵*/
```

```
public int[][] checkMatrix(int n) {
	ArrayList<Integer> array=new ArrayList<Integer>();
	int m=(int )(Math.pow(2,n)-1);
	int [][] H=new int [n][m];

	for (int i=0;i<n;i++)
		for (int j=0;j<n;j++)
			H[i][j]=0;

	for (int i=0;i<n;i++)
		array.add((int )(Math.pow(2,i)));
	int p=0,q=0,mm=1;
	String b=new String();
	for (;q<m-n;) {
		b=Integer.toBinaryString(mm);
		if (!array.contains(mm)) {
			for (int j=n-1;j>=0;j--) {
				if (p==b.length()) break;
				else {
H[j][q]=Integer.parseInt(b.substring(b.length()-(p+1),b.length()-p));
					p++;
				}
			}
			q++;
			p=0;
		}
		mm++;
	}
	for (int i=0;i<n;i++) {
		for (int j=0;j<n;j++)
			if (i==j)
				H[i][j+(int)(Math.pow(2,n)-1-n)]=1;
	}
	return H;
}

/*生成矩阵*/
public int[][] creatMatrix(int H[][],int n){
	int i,j,m=(int)(Math.pow(2, n)-1);
	int mm=(int) (Math.pow(2, n)-1-n);
	int[][] G=new int[mm][m];

	for (i=0;i<mm;i++)
		for (j=0;j<mm;j++)
			if (i==j) G[i][j]=1;
	for (i=0;i<n;i++)
		for (j=0;j<mm;j++)
			G[j][i+mm]=H[i][j];
	return G;
```

```
}

/*Hamming 码*/
public int[] HammingCoding(int[][] g,ArrayList<Integer> array) {
	int i,j,num;
	int[] code=new int[g[1].length];
	for (i=0;i<g[1].length;i++) {
		num=0;
		for (j=0;j<g.length;j++)
			num+=array.get(j)*g[j][i];
		if ((num & 1)==1) code[i]=1;
		else code[i]=0;
	}
	return code;
}


/*检错纠错*/
public void checkCorrectError(int h[][],ArrayList<Integer> array) {
	int i,j,num;
	int [] HX=new int[h.length];
	for (i=0;i<h.length;i++) {
		num=0;
		for (j=0;j<array.size();j++)
			num+=h[i][j]*array.get(j);
		if ((num & 1)==1) HX[i]=1;
		else HX[i]=0;
	}
	print("H*X=[");
	for (i=0;i<HX.length;i++) {
		print (HX[i]);
		if (i!=HX.length-1) print(",");
	}
	print("]\n");

	boolean flag=true;
	for (i=1;i<HX.length;i++) {
		if (HX[i]!=0) {
			flag=false;
			break;
		}
	}
	if (! flag) {
		int m=0;
		m=researchIndex(h,HX);
		if (m!=0) {
			print("接收的信息出错，出错位置为第"+m+"位.\n");
			print("纠错前的信息为:");
			for (i=0;i<array.size();i++)
```

```
                        print(array.get(i));
                    /*纠错*/
                    m=m-1;
                    array.set(m, array.get(m)^1);
                    print("\n 纠错后的信息为: ");
                    for (i=0;i<array.size();i++) print(array.get(i));
                    print("\n");
                }
            }
            else {
                print("接收的信息正确，信息为: ");
                for (i=0;i<array.size();i++) print(array.get(i));
                print("\n");
            }
        }


        public int researchIndex(int[][] h, int[] hX) {
            int i,j,num;
            for (i=0;i<h[1].length;i++) {
                num =0;
                for (j=0;j<h.length;j++) {
                    if (hX[j]==h[j][i]) num++;
                }

                if (num==h.length) return i+1;
            }
            return 0;
        }



        public void print(int str) {
            System.out.print(str);
        }
        public void print(String str) {
            System.out.print(str);
        }


}
```

TestHamming.java

```
package IT7;

import java.util.ArrayList;
```

```
import java.util.Scanner;

public class TestHamming {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Hamming HM=new Hamming();

		ArrayList<Integer> array1=new ArrayList<Integer>();
		ArrayList<Integer> array2=new ArrayList<Integer>();

		int in,k;

		int[][] H;//校验矩阵
		int[][] G;//生成矩阵
		String input;
		boolean flag1=true,flag2=true,flag3=true,flag4=true;
		while (1==1) {
			k=0;
			array1.clear();
			array2.clear();
			HM.print("继续? Y/N:");
			Scanner s=new Scanner(System.in);
			input=s.next();

			if (input.equals("N")) break;
			HM.print("请输入待编码的二进制字符串,以 2 结束");
			in=s.nextInt();
			if (!(in==1 || in ==0)) HM.print("输入错误，请重新输入\n");
			while (in!=2 &&(in==1||in==0)) {
				array1.add(in);
				in=s.nextInt();
				if ((in==1 || in ==0 || in==2)==false)
					flag1=false;
			}

			if (!flag1) HM.print("输入错误，请重新输入\n");

			for (int i=0;;i++) {
				if (Math.pow(2,k)-k==array1.size()+1) {
					flag3=true;
					break;
				}
				if (Math.pow(2,k)-k>array1.size()+1) {
					flag3=true;
					break;
				}
				k++;
			}
```

```
                if (flag3==false) HM.print("字符个数输入出错，请重新输
入.\n");



                if (flag3) {
                      H=HM.checkMatrix(k);
                      G=HM.creatMatrix(H, k);

                      HM.print("校验矩阵为：\n");

                      for (int i=0;i<H.length;i++) {
                            for (int j=0;j<H[i].length;j++)
                                  System.out.print(" "+H[i][j]);
                            System.out.println();

                      }


                      HM.print("生成矩阵为：\n");

                      for (int i=0;i<G.length;i++) {
                            for (int j=0;j<G[i].length;j++)
                                  System.out.print(" "+G[i][j]);
                            System.out.println();
                      }

                      HM.print("Hamming 编码为\n");
                      int[] code=HM.HammingCoding(G, array1);
                      for (int i=0;i<code.length;i++)
                            HM.print(code[i]+" ");

                      int mm=(int)(Math.pow(2, k)-1);
                      while (flag4) {
                            HM.print("\n 请输入"+mm+"位接收方接收的字符：");
                            for (int i=0;i<mm;i++) {
                                  in=s.nextInt();
                                  if ((in==1 ||  in==0)==false) {
                                        HM.print("输入错误，请重新输入");
                                        break;
                                  }

                                  else {
                                        if (i==mm-1) flag4=false;
                                        array2.add(in);
                                  }
                            }
```

```
                    }
                    flag4=true;
                    HM.checkCorrectError(H, array2);
                }
            }
        }

}
```

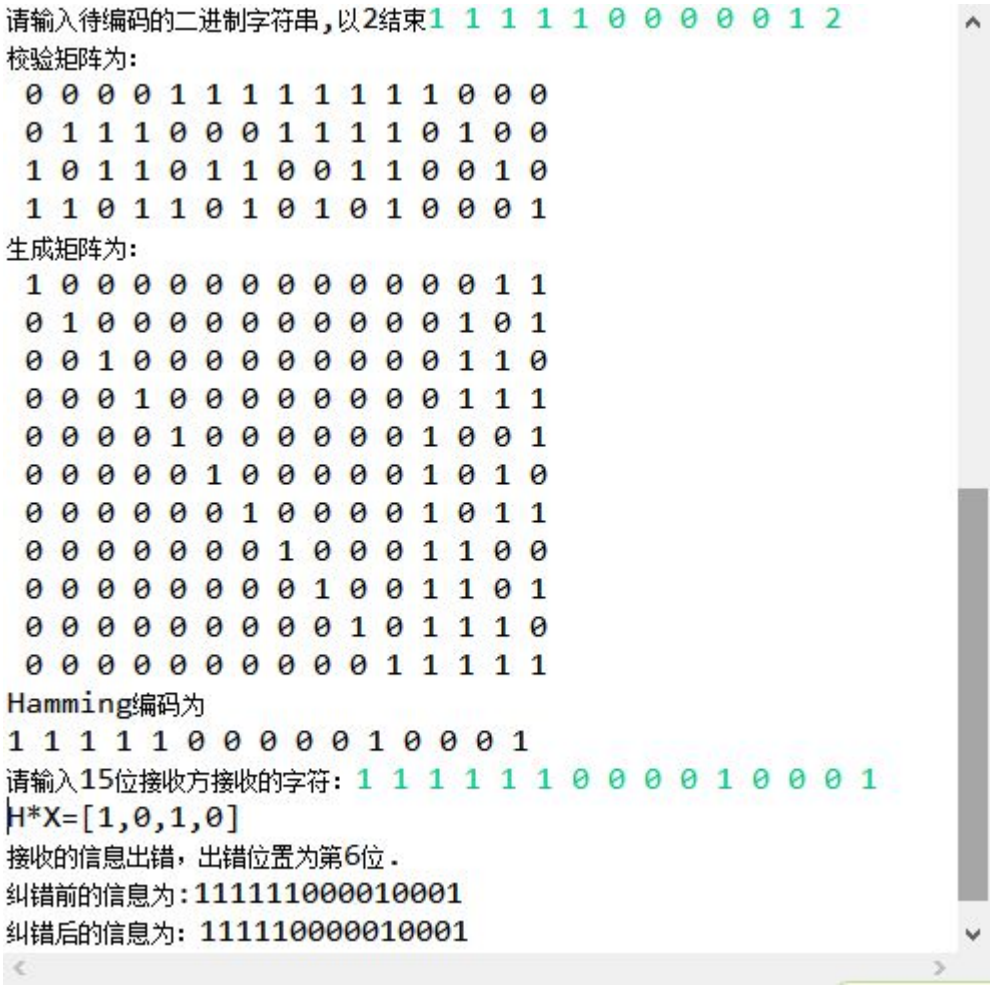
测试结果：

①当正确接收时

```
继续? Y/N:Y
请输入待编码的二进制字符串,以2结束1 1 1 1 1 0 0 0 0 0 1 2
校验矩阵为:
 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0
 0 1 1 1 0 0 0 1 1 1 1 0 1 0 0
 1 0 1 1 0 1 1 0 0 1 1 0 0 1 0
 1 1 0 1 1 0 1 0 1 0 1 0 0 0 1
生成矩阵为:
 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1
 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1
 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0
 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1
 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1
 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0
 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1
 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0
 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1
 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0
 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1
Hamming编码为
1 1 1 1 1 0 0 0 0 0 1 0 0 0 1
请输入15位接收方接收的字符: 1 1 1 1 1 0 0 0 0 0 1 0 0 0 1
H*X=[0,0,0,0]
接收的信息正确，信息为: 111110000010001
继续? Y/N:
```

② 当第一位接收出错时：

③当第 6 接收出错时



④当第 1,2 位接收出错时

检测出有出错，但是没有正确给出出错位数，纠错也失败。

五、实验体会

（请认真填写自己的真实体会）

经过实验测试，发现如下

1.当接收方正确接收时，HX=[0 0 0 0]，正确检测接收方的正确；

2.设置一位出错，程序找到出错位置，且能正确纠错；

3.设置两位出错，程序检测出有出错，但是没有正确给出出错位数，纠错也失败。

4.自然，三位以上的出错检错纠错都将失败。

六、参考文献

1. Thomas M. Cover, Joy A. Thomas. Elements of Information Theory (2nd Edition) [M]. John Wiley & Sons, Inc. Chapter 7

2.（如有其它参考文献，请列出）