INFORMATION THEORY

The Implementation of Huffman Coder

2016-1-7

莫帮杰
2013301200227

信息论与编码

——Huffman 编码器的实现

一、 实验目的

1、理解和掌握Huffman编码的基本原理和方法；

2、给出信源符号的概率统计分布，并计算信源熵；

3、给出两种以上单信源符号Huffman码表；

4、计算每信源的平均字长，并与信源熵比较；

5、计算编码效率，分析码长方差对实验系统的影响。

二、 实验原理

1、 二元霍夫曼编码的基本原理

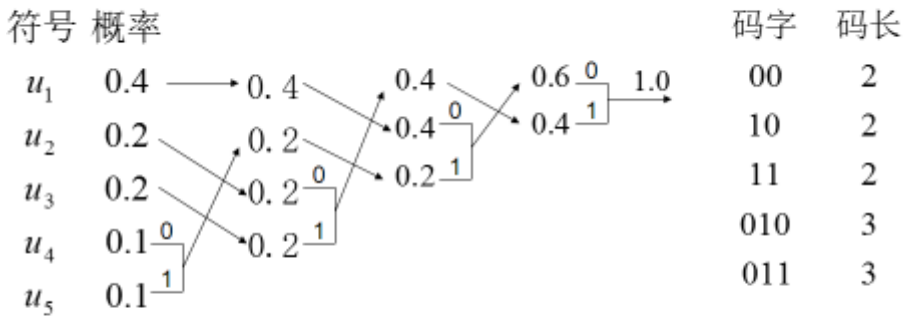
（1）将q个信源符号按照概率分布P(s_i)的大小顺序递增（或递减）排列：

P_1<=P_2<=P_3<=…<=P_q；

（2）用码符号0和1分别给概率最小的两个信源符号编码，并将这两个符号合并成一个新符号，新符号的概率即为这两个符号的概率之和，从而得到只包含q-1个符号的新信源S_i；

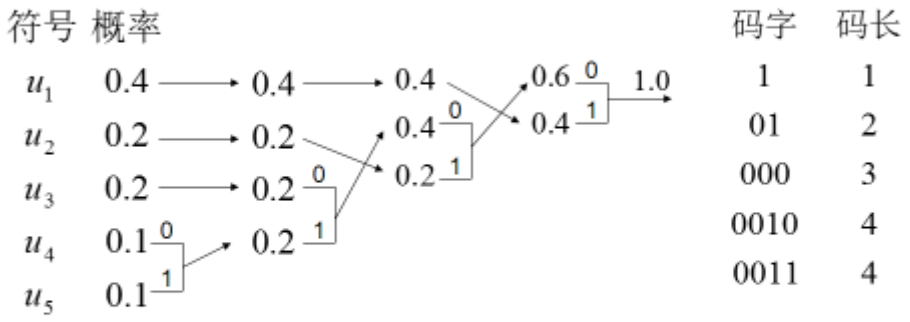
（3）重复步骤（1）~（2）,直至信源不能再缩减；

（4）从编码路径向前返回，所得到的路径序列即为对应符号的码字。



(a)

图5-2 例5-2 两种霍夫曼编码



(b)

图5-2 例5-2 两种霍夫曼编码

由于码符号分配的任意性和符号合并后概率与其他符号相同时排序方法的不同，同一信源的霍夫曼编码往往并不唯一。但是，针对第二种情况，编码时如尽量把新生成的码符号概率排在最前面将有利于减小码长方差，编出来的码也就跟接近于等长码，这在后面的实验结果当中会有所体现。

2、r元霍夫曼编码的基本原理

（1）将q个信源符号按照概率分布$P(s_i)$的大小顺序递增（或递减）排列：

P_1<=P_2<=P_3<=…<=P_q；

（2）用r个码符号a_0、a_1、a_2…a_{r-1}分别给概率最小的r个信源符号编码，并将这r个符号合并成一个新符号，新符号的概率即为这r个符号的概率之和，从而得到只包含q-r+1个符号的新信源S_i；

（3）重复步骤（1）~（2），直至信源不能再缩减；

（4）从编码路径向前返回，所得到的路径序列即为对应符号的码字。

需要注意的是，由于r元霍夫曼编码每次缩减r-1个符号，为保证缩减n次后信源符号个数恰好等于r，在编码前需检查初始信源符号个数q（q>=2）是否满足q=(r-1)*n+r，如果不满足，则填充最少的0使满足。特殊的，在r=2，即二元编码的条件下，q=(r-1)*n+r总是成立的。

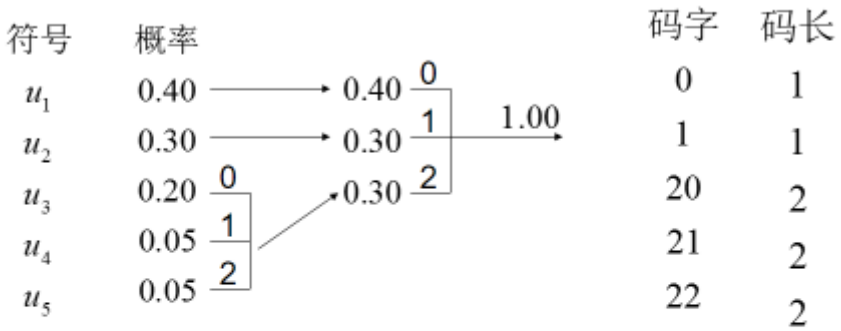


图5-4 三元霍夫曼编码

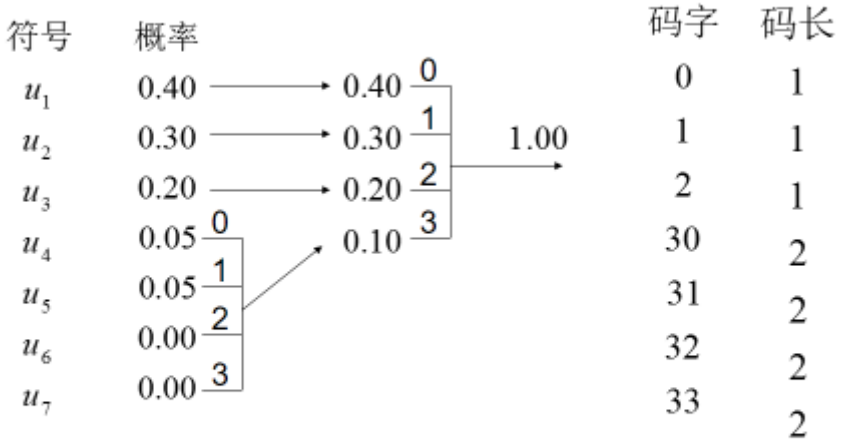


图5-5 四元霍夫曼编码

3. 平均码长：$\mathrm{L}=\sum P(s_i)*l_i$ （单位为：码符号/信源符号）

其中，$P(s_i)$为信源S在q个信源符号中出现的概率，l_i为编码后信源S的第i个码字的码长。

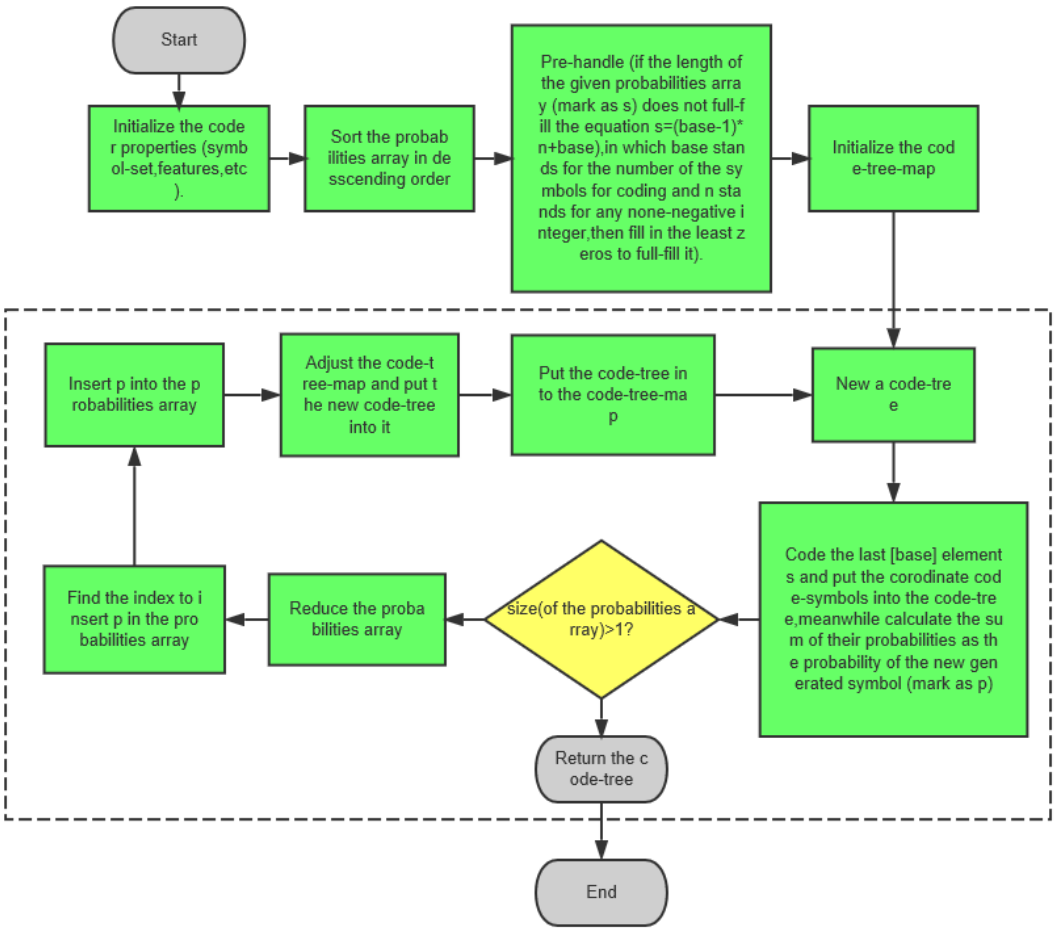
4. 信息熵：$\mathrm{H(S)}=-\sum P(s_i)*log\ P(s_i)$ （单位为：比特/信源符号）

其中，$P(s_i)$为信源S在q个信源符号中出现的概率。

5. 编码效率：$\eta=\mathrm{H(S)/L}$

其中，H(S)为信息熵，L为平均码长。

三、 算法设计



Created by Bangjie Mo

四、 JAVA 关键代码

```
/**
 * 迭代编码，由下至上构建码树
 * @param probabilities 信源的概率分布
 * @param codeTreeMap 码树图
 * @return 码树
 */
private CodeTree<T> code(double[] probabilities, CodeTreeMap<T> codeTreeMap) {
    //每一次迭代都生成新的码树
```

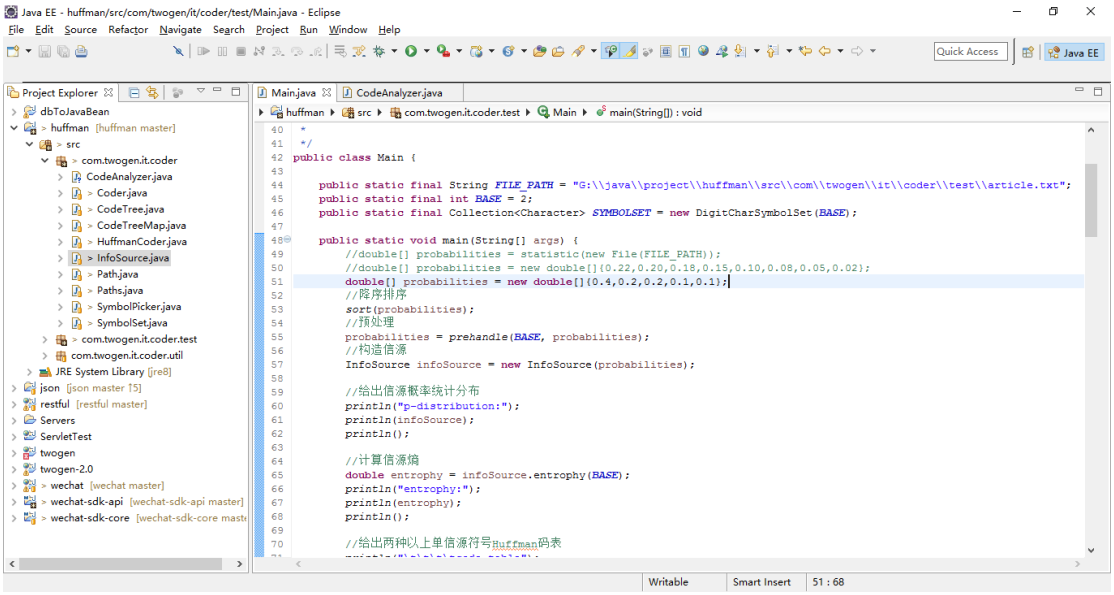
```
CodeTree<T> codeTree = new CodeTree<T>();
//符号挑选器
SymbolPicker<T> symbolPicker = new SymbolPicker<T>(arbitrary ,symbolSet);
//新信源符号的概率
double probability = 0;
for (int i = probabilities.length-1; i > probabilities.length-base-1; i--) {
    //累加生成新信源符号的概率
    probability += probabilities[i];
    //下一个符号
    T symbol = symbolPicker.next();
    //从码树图中取得游离的历史码树，放入新的码树中构建，同时将其移除（忘记）
    codeTree.put(symbol,codeTreeMap.remove(i));
}
//迭代直到信源大小不能再缩减为止
if (probabilities.length == base) {
    return codeTree;
}
//缩减信源
probabilities = reduce(probabilities);
//找到新信源符号待插入位置
int index = indexToInsert(probability, probabilities);
//插入新信源符号
insert(probabilities, probability, index);
//调整码树图
adjustCodeTreeMap(codeTreeMap, index);
//记住新生成的码树
codeTreeMap.put(index, codeTree);
//继续下一轮编码
return code(probabilities, codeTreeMap);
```

说明：以上只是代码中比较核心的部分，完整的可运行程序请查看附件中的 JAVA 代码包。

五、 实验过程

1、 二元 huffman 编码验证性实验

用 Eclipse 打开源码，在测试类中将 BASE 设置为 2，输入信源分布 p=[0.4,0.2,0.2,0.1,0.1]，构造具有随机编码、提供优化特性的 huffman 编码器，设置循环 5 次编码，如图：



配置好后点击运行 JAVA 程序，在控制台输出以下测试内容：

p-distribution:
[0.4,0.2,0.2,0.1,0.1]

entrophy:
2.1219280948873624

code-table(0)

probability	code-word	code-word length
0.4	00	2
0.2	10	2
0.2	11	2
0.1	010	3
0.1	011	3
average code-word length:2.2		
code-word length variance:0.16		
code efficiency:96.45127704033466%		

code-table(1)

probability	code-word	code-word length
0.4	01	2
0.2	10	2
0.2	11	2
0.1	000	3
0.1	001	3
average code-word length:2.2		
code-word length variance:0.16		
code efficiency:96.45127704033466%		

code-table(2)

probability	code-word	code-word length
0.4	00	2
0.2	01	2
0.2	10	2
0.1	110	3
0.1	111	3
average code-word length:2.2		
code-word length variance:0.16		
code efficiency:96.45127704033466%		

code-table(3)

probability	code-word	code-word length
0.4	01	2
0.2	10	2
0.2	11	2
0.1	000	3
0.1	001	3
average code-word length:2.2		
code-word length variance:0.16		
code efficiency:96.45127704033466%		

code-table(4)

probability	code-word	code-word length
0.4	00	2
0.2	10	2
0.2	11	2
0.1	010	3
0.1	011	3
average code-word length:2.2		
code-word length variance:0.16		
code efficiency:96.45127704033466%		

可见有优化的随机 huffman 编码的编码结果可能不尽相同，但他们的平均码长、码长方差和编码效率是一致的，所以可以把他们看作等价码。

下面再来看一下无优化的随机二元 huffman 编码。将编码器构造函数的第三个参数改为 false，其他不变，再次运行编码程序，在控制台有新的输出：

```
p-distribution:
[0.4,0.2,0.2,0.1,0.1]

entrophy:
2.1219280948873624

                              code-table(0)
 ------------------------------------------------------------------------------
|       probability     |       code-word       |       code-word length        |
 ------------------------------------------------------------------------------
|       0.4             |       1               |       1                       |
 ------------------------------------------------------------------------------
|       0.2             |       00              |       2                       |
 ------------------------------------------------------------------------------
|       0.2             |       011             |       3                       |
 ------------------------------------------------------------------------------
|       0.1             |       0100            |       4                       |
 ------------------------------------------------------------------------------
|       0.1             |       0101            |       4                       |
 ------------------------------------------------------------------------------
|       average code-word length:2.2                                            |
 ------------------------------------------------------------------------------
|       code-word length variance:1.3599999999999999
 ------------------------------------------------------------------------------
|       code efficiency:96.45127704033466%                                      |
 ------------------------------------------------------------------------------

                              code-table(1)
 ------------------------------------------------------------------------------
|       probability     |       code-word       |       code-word length        |
 ------------------------------------------------------------------------------
|       0.4             |       1               |       1                       |
 ------------------------------------------------------------------------------
|       0.2             |       01              |       2                       |
 ------------------------------------------------------------------------------
|       0.2             |       000             |       3                       |
 ------------------------------------------------------------------------------
|       0.1             |       0010            |       4                       |
 ------------------------------------------------------------------------------
|       0.1             |       0011            |       4                       |
 ------------------------------------------------------------------------------
|       average code-word length:2.2                                            |
 ------------------------------------------------------------------------------
|       code-word length variance:1.3599999999999999
 ------------------------------------------------------------------------------
|       code efficiency:96.45127704033466%                                      |
 ------------------------------------------------------------------------------
```

code-table(2)

probability	code-word	code-word length
0.4	1	1
0.2	01	2
0.2	001	3
0.1	0000	4
0.1	0001	4
average code-word length:2.2		
code-word length variance:1.3599999999999999		
code efficiency:96.45127704033466%		

code-table(3)

probability	code-word	code-word length
0.4	0	1
0.2	10	2
0.2	111	3
0.1	1100	4
0.1	1101	4
average code-word length:2.2		
code-word length variance:1.3599999999999999		
code efficiency:96.45127704033466%		

code-table(4)

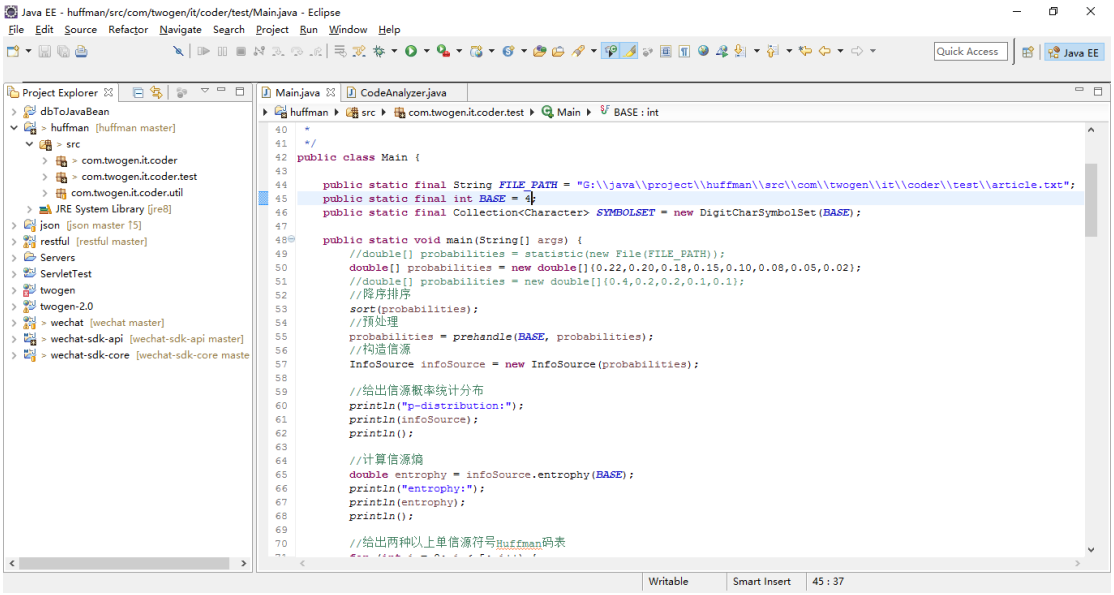
probability	code-word	code-word length
0.4	0	1
0.2	11	2
0.2	101	3
0.1	1000	4
0.1	1001	4
average code-word length:2.2		
code-word length variance:1.3599999999999999		
code efficiency:96.45127704033466%		

从输出结果可以看到，无优化的随机 huffman 编码的编码结果也不尽相同，但他们的平均码长、码长方差和编码效率也是一致的，所以也可以把他们看作等价码。

对比有优化和无优化两种码型可见，有优化时编出来的 huffman 码方差更小、更接近于等长码，原因是有优化时程序会尽量将合并后的码符号放在最前面，从而减少重复编码的次数，使短码得到充分利用。故实际应用时应采用有优化的编码方式。

2、四元 huffman 编码验证性实验

在测试类中将 BASE 设置为 4，输入信源分布 p=[0.22,0.20,0.18,0.15,0.10,0.08,0.05,0.02]，构造具有随机编码、提供优化特性的 huffman 编码器，设置循环 5 次编码，如图：



运行程序，在 JAVA 控制台输出：

```
p-distribution:
[0.22,0.2,0.18,0.15,0.1,0.08,0.05,0.02,0.0,0.0]

entrophy:
1.3767431554315241
```

code-table(0)

probability	code-word	code-word length
0.22	1	1
0.2	2	1
0.18	3	1
0.15	01	2
0.1	02	2
0.08	03	2
0.05	000	3
0.02	001	3
0.0	002	3
0.0	003	3
average code-word length:1.4700000000000002		
code-word length variance:0.3891		
code efficiency:93.65599696813088%		

code-table(1)

probability	code-word	code-word length
0.22	0	1
0.2	1	1
0.18	3	1
0.15	20	2
0.1	21	2
0.08	22	2
0.05	230	3
0.02	231	3
0.0	232	3
0.0	233	3
average code-word length:1.4700000000000002		
code-word length variance:0.3891		
code efficiency:93.65599696813088%		

code-table(2)

probability	code-word	code-word length
0.22	0	1
0.2	2	1
0.18	3	1
0.15	10	2
0.1	11	2
0.08	12	2
0.05	130	3
0.02	131	3
0.0	132	3
0.0	133	3
average code-word length:1.4700000000000002		
code-word length variance:0.3891		
code efficiency:93.65599696813088%		

code-table(3)

probability	code-word	code-word length
0.22	0	1
0.2	1	1
0.18	3	1
0.15	20	2
0.1	21	2
0.08	22	2
0.05	230	3
0.02	231	3
0.0	232	3
0.0	233	3
average code-word length:1.4700000000000002		
code-word length variance:0.3891		
code efficiency:93.65599696813088%		

code-table(4)

probability	code-word	code-word length
0.22	0	1
0.2	1	1
0.18	3	1
0.15	20	2
0.1	21	2
0.08	22	2
0.05	230	3
0.02	231	3
0.0	232	3
0.0	233	3
average code-word length:1.4700000000000002		
code-word length variance:0.3891		
code efficiency:93.65599696813088%		

将 huffman 编码器构造函数的第三个参数改为 false，重新运行程序，输出结果如下：

```
p-distribution:
[0.22,0.2,0.18,0.15,0.1,0.08,0.05,0.02,0.0,0.0]

entrophy:
1.3767431554315241
```

code-table(0)

probability	code-word	code-word length
0.22	1	1
0.2	2	1
0.18	3	1
0.15	00	2
0.1	01	2
0.08	03	2
0.05	020	3
0.02	021	3
0.0	022	3
0.0	023	3
average code-word length:1.4700000000000002		
code-word length variance:0.3891		
code efficiency:93.655996968130 88%		

code-table(1)

probability	code-word	code-word length
0.22	0	1
0.2	1	1
0.18	3	1
0.15	20	2
0.1	21	2
0.08	23	2
0.05	220	3
0.02	221	3
0.0	222	3
0.0	223	3
average code-word length:1.4700000000000002		
code-word length variance:0.3891		
code efficiency:93.655996968130 88%		

code-table(2)

probability	code-word	code-word length
0.22	0	1
0.2	1	1
0.18	2	1
0.15	31	2
0.1	32	2
0.08	33	2
0.05	300	3
0.02	301	3
0.0	302	3
0.0	303	3
average code-word length:1.4700000000000002		
code-word length variance:0.3891		
code efficiency:93.655996968813088%		

code-table(3)

probability	code-word	code-word length
0.22	0	1
0.2	1	1
0.18	3	1
0.15	21	2
0.1	22	2
0.08	23	2
0.05	200	3
0.02	201	3
0.0	202	3
0.0	203	3
average code-word length:1.4700000000000002		
code-word length variance:0.3891		
code efficiency:93.655996968813088%		

code-table(4)

probability	code-word	code-word length
0.22	1	1
0.2	2	1
0.18	3	1
0.15	00	2
0.1	01	2
0.08	02	2
0.05	030	3
0.02	031	3
0.0	032	3
0.0	033	3
average code-word length:1.4700000000000002		
code-word length variance:0.3891		
code efficiency:93.655996968813088%		

3、 找一篇 1000 个单词以上的英文文章，只考虑 26 个字母和空格(标点符号当空格处理)，进行 Huffman 统计编解码实验。以下为实验结果：

original source:

My father was a self-taught mandolin player. He was one of the best string instrument players in our town. He could not read music, but if he heard a tune a few times, he could play it. When he was younger, he was a member of a small country music band. They would play at local dances and on a few occasions would play for the local radio station. He often told us how he had auditioned and earned a position in a band that featured Patsy Cline as their lead singer. He told the family that after he was hired he never went back. Dad was a very religious man. He stated that there was a lot of drinking and cursing the day of his audition and he did not want to be around that type of environment. Occasionally, Dad would get out his mandolin and play for the family. We three children: Trisha, Monte and I, George Jr., would often sing along. Songs such as the Tennessee Waltz, Harbor Lights and around Christmas time, the well-known rendition of Silver Bells. "Silver Bells, Silver Bells, its Christmas time in the city" would

ring throughout the house. One of Dad's favorite hymns was "The Old Rugged Cross". We learned the words to the hymn when we were very young, and would sing it with Dad when he would play and sing. Another song that was often shared in our house was a song that accompanied the Walt Disney series: Davey Crockett. Dad only had to hear the song twice before he learned it well enough to play it. "Davey, Davey Crockett, King of the Wild Frontier" was a favorite song for the family. He knew we enjoyed the song and the program and would often get out the mandolin after the program was over. I could never get over how he could play the songs so well after only hearing them a few times. I loved to sing, but I never learned how to play the mandolin. This is something I regret to this day. Dad loved to play the mandolin for his family he knew we enjoyed singing, and hearing him play. He was like that. If he could give pleasure to others, he would, especially his family. He was always there, sacrificing his time and efforts to see that his family had enough in their life. I had to mature into a man and have children of my own before I realized how much he had sacrificed. I joined the United States Air Force in January of 1962. Whenever I would come home on leave, I would ask Dad to play the mandolin. Nobody played the mandolin like my father. He could touch your soul with the tones that came out of that old mandolin. He seemed to shine when he was playing. You could see his pride in his ability to play so well for his family. When Dad was younger, he worked for his father on the farm. His father was a farmer and sharecropped a farm for the man who owned the property. In 1950, our family moved from the farm. Dad had gained employment at the local limestone quarry. When the quarry closed in August of 1957, he had to seek other employment. He worked for Owens Yacht Company in Dundalk, Maryland and for Todd Steel in Point of Rocks, Maryland. While working at Todd Steel, he was involved in an accident. His job was to roll angle iron onto a conveyor so that the welders farther up the production line would have it to complete their job. On this particular day Dad got the third index finger of his left hand mashed between two pieces of steel. The doctor who operated on the finger could not save it, and Dad ended up having the tip of the finger amputated. He didn't lose enough of the finger where it would stop him picking up anything, but it did impact his ability to play the mandolin. After the accident, Dad was reluctant to play the mandolin. He felt that he could not play

as well as he had before the accident. When I came home on leave and asked him to play he would make excuses for why he couldn't play. Eventually, we would wear him down and he would say "Okay, but remember, I can't hold down on the strings the way I used to" or "Since the accident to this finger I can't play as good". For the family it didn't make any difference that Dad couldn't play as well. We were just glad that he would play. When he played the old mandolin it would carry us back to a cheerful, happier time in our lives. "Davey, Davey Crockett, King of the Wild Frontier", would again be heard in the little town of Bakerton, West Virginia. In August of 1993 my father was diagnosed with inoperable lung cancer. He chose not to receive chemotherapy treatments so that he could live out the rest of his life in dignity. About a week before his death, we asked Dad if he would play the mandolin for us. He made excuses but said "okay". He knew it would probably be the last time he would play for us. He tuned up the old mandolin and played a few notes. When I looked around, there was not a dry eye in the family. We saw before us a quiet humble man with an inner strength that comes from knowing God, and living with him in one's life. Dad would never play the mandolin for us again. We felt at the time that he wouldn't have enough strength to play, and that makes the memory of that day even stronger. Dad was doing something he had done all his life, giving. As sick as he was, he was still pleasing others. Dad sure could play that Mandolin!

statistic:

{ =1149.0, A=7.0, B=4.0, C=8.0, D=24.0, E=1.0, F=4.0, G=2.0, H=20.0, I=18.0, J=2.0, K=2.0, L=1.0, M=5.0, N=1.0, O=6.0, P=2.0, R=2.0, S=8.0, T=8.0, U=1.0, V=1.0, W=18.0, Y=2.0, a=344.0, b=32.0, c=84.0, d=214.0, e=478.0, f=100.0, g=80.0, h=241.0, i=242.0, j=6.0, k=32.0, l=210.0, m=98.0, n=279.0, o=306.0, p=66.0, q=3.0, r=194.0, s=187.0, t=327.0, u=118.0, v=41.0, w=109.0, x=3.0, y=103.0, z=2.0}

p-distribution:

[0.22117420596727622,0.09201154956689124,0.06621751684311838,0.0629451395572666,0.05890279114533205,0.05370548604427334,0.0465832531280077,0.04639076034648701,0.0411934552454283,0.04042348411934552,0.03734359961501444

,0.03599615014436958,0.02271414821944177,0.020981713185755535,0.019826756496631376,0.019249278152069296,0.01886429258902791,0.01616939364773821,0.015399422521655439,0.012704523580365737,0.007892204042348411,0.006159769008662175,0.006159769008662175,0.004619826756496631,0.0038498556304138597,0.0034648700673724736,0.0034648700673724736,0.0015399422521655437,0.0015399422521655437,0.0015399422521655437,0.0013474494706448511,0.0011549566891241579,0.0011549566891241579,9.624639076034649E-4,7.699711260827719E-4,7.699711260827719E-4,5.774783445620789E-4,5.774783445620789E-4,3.8498556304138594E-4,3.8498556304138594E-4,3.8498556304138594E-4,3.8498556304138594E-4,3.8498556304138594E-4,3.8498556304138594E-4,3.8498556304138594E-4,1.9249278152069297E-4,1.9249278152069297E-4,1.9249278152069297E-4,1.9249278152069297E-4,1.9249278152069297E-4,1.9249278152069297E-4]

entrophy:

4.181669518192732

code-table(0)

probability	code-word	code-word length
0.22117420596727622	11	2
0.09201154956689124	0000	4
0.06621751684311838	0001	4
0.0629451395572666	0011	4
0.05890279114533205	0101	4

| 0.05370548604427334 | 1000 | 4

| 0.0465832531280077 | 1010 | 4

| 0.04639076034648701 | 1011 | 4

| 0.0411934552454283 | 01000 | 5

| 0.04042348411934552 | 01100 | 5

| 0.03734359961501444 | 01110 | 5

| 0.03599615014436958 | 01111 | 5

| 0.02271414821944177 | 10010 | 5

| 0.020981713185755535 | 001000 | 6

| 0.019826756496631376 | 001001 | 6

| 0.019249278152069296 | 001011 | 6

| 0.01886429258902791 | 010011 | 6

| 0.01616939364773821 | 011010 | 6

| 0.015399422521655439 | 011011 | 6

| 0.012704523580365737 | 100111 | 6

| 0.007892204042348411 | 0010101 | 7

| 0.006159769008662175 | 00101000 | 8

| 0.006159769008662175 | 01001000 | 8

| 0.0046198267564966315 | 01001001 | 8

| 0.0038498556304138597 | 01001011 | 8

| 0.0034648700673724736 | 10011010 | 8

| 0.0034648700673724736 | 10011011 | 8

| 0.0015399422521655437 | 100110000 | 9

| 0.0015399422521655437 | 100110001 | 9

| 0.0015399422521655437 | 0010100100 | 10

| 0.001347449470644851 | 0010100110 | 10

| 0.0011549566891241579 | 0010100111 | 10

| 0.0011549566891241579 | 0100101000 | 10

| 9.624639076034649E-4 | 0100101011 | 10

| 7.699711260827719E-4 | 1001100111 | 10

| 7.699711260827719E-4 | 01001010011 | 11

| 5.774783445620789E-4 | 01001010100 | 11

| 5.774783445620789E-4 | 01001010101 | 11

| 3.8498556304138594E-4 | 10011001000 | 11

| 3.8498556304138594E-4 | 10011001001 | 11

| 3.8498556304138594E-4 | 10011001010 | 11

| 3.8498556304138594E-4 | 10011001011 | 11

| 3.8498556304138594E-4 | 10011001100 | 11

| 3.8498556304138594E-4 | 10011001101 | 11

| 3.8498556304138594E-4 | 001010010100 | 12

| 1.9249278152069297E-4 | 001010010101 | 12

| 1.9249278152069297E-4 | 001010010110 | 12

| 1.9249278152069297E-4 | 001010010111 | 12

| 1.9249278152069297E-4 | 010010100100 | 12

| 1.9249278152069297E-4 | 010010100101 | 12

| average code-word length:4.213666987487971

| code-word length variance:2.64424054742798

| code efficiency:99.24062652814632%

输出结果太长，放在此处可能不方便查阅，故另外将其放在附件中，可点击查阅：

article-huffuman-coding-result.txt

七、 结语

本文通过 JAVA 平台实现了 huffman 编码的计算机算法实现，研究和展示了 huffman 编码的过程和特性，并验证了 huffman 编码的高效性。唯一不足的是还没有时间来实现 shanon 编码和 fano 编码，并与 huffman 编码进行比较，待他日有时间再来讨论。