编码PK实验报告

一、实验概述

基于课堂上的分组对战，完成组间的测试与PK，分别对一段中英文(3000字左右)的文本进行Shannoncodes和 Q-ary Huffman codes的编解码。

二、编码算法原理

（一）Shannon 编码

基于信源符号概率分布，按公式 $l_i = \lceil -\log_2 p_i \rceil$（$l_i$ 为码长，p_i 为符号概率 ）确定各符号码长，再通过累积概率分配二进制编码，实现信源压缩，理论上接近熵值下限。

（二）Huffman 编码

构建优先队列（最小堆）存储符号概率，反复合并概率最小的节点生成 Huffman 树，依据树结构（左 0 右 1 等规则）为符号分配变长编码，无失真压缩，平均码长接近熵。

（三）Q - ary Huffman 编码

扩展 Huffman 编码思想，采用 Q 叉树（$Q \geq 2$ 整数 ），合并 Q 个概率最小节点构建编码树，为符号分配 Q 进制编码，适配不同编码基数需求，在多进制场景优化压缩效率 。

三、测试环境与内容

（一）环境

编程语言：Python
运行终端：基于 Anaconda 环境的命令行
测试文本：中英文各 3000 字左右。

（二）内容

1. 自研代码测试：对自身开发的 Shannon、Huffman 编解码脚本，用中英文文本测试，输出熵（反映信源不确定性 ）、平均码长（编码效率关键指标 ）、编码效率（$\eta = \frac{H(X)}{\bar{L}} \times 100\%$ ，$H(X)$ 为熵，$\bar{L}$ 为平均码长 ）、解码准确率（验证还原能力 ）及时间。
2. 组间 PK 测试：交换测试信源，用对手代码编解码，记录同指标，对比差异。

四、测试结果与分析

（一）Shannon 编码

1. 本组代码-多组文本

编码类型	文本类型	熵 (bits/symbol)	平均码长 (bits/symbol)	编码效率	解码准确率	解码时间 (seconds)
Shannon	本组	4.4790	4.6608	96.10%	100.00%	0.0008

编码类型	文本类型	熵 (bits/symbol)	平均码长 (bits/symbol)	编码效率	解码准确率	解码时间 (seconds)
	英文					
Shannon	本组中文	8.0785	8.2584	97.82%	100.00%	0.0006
Shannon	5组中文	7.9203	8.1481	97.21%	100.00%	0.0018
Shannon	5组英文	4.2748	4.4762	95.50%	100.00%	0.0010
Shannon	8组中文	8.1705	8.4730	96.43%	100.00%	0.0010
Shannon	8组英文	4.3885	4.5585	96.27%	100.00%	0.0036

分析：解码准确率均 100%，说明算法逻辑正确。英文文本因字符集小、概率分布相对集中，编码效率整体略高（如英文 1 达 96.10% ）；中文符号多、分布分散，平均码长接近熵值难度大，效率稍低。编码效率与信源概率分布相关，分布越集中，越接近熵下限，效率越高 。

2. 多组代码-本组文本

测试组	语言	信源熵 (bits/symbol)	平均码长 (bits/symbol)	编码效率	压缩率 (空间节省)	解码耗时 (ms)
组5	中文	8.0812	8.4022	96.18%	-	-
	英文	4.6123	5.0929	90.56%	-	-
组8	中文	8.0812	8.4022	96.18%	64.53%	2.3390
	英文	4.6123	5.0929	90.56%	37.01%	1.8720

（二）Huffman 编码结果

以中文、英文各一段测试（Q = 2,3,4,5 ），熵固定（如中文熵 8.0812、英文 4.4790 ），随 Q 增大：

- **平均码长**：Q 进制符号数增多，平均码长（以对应进制符号数计 ）减小，如中文 Q = 2 时平均码长 8.1043（base - 2 符号 ），Q = 5 时降为 3.8946（base - 5 符号 ）；
- **编码效率**：先升后降，Q 适配信源概率分布时效率高（如中文 Q = 2 达 99.71% ），Q 过大（如中文 Q = 5 降至 89.36% ），因合并节点规则限制，编码树构建难最优，效率下滑；
- **解码准确率**：始终 100%，算法还原逻辑可靠；
- **解码时间**：随 Q 增大略有波动，Q 增大时编码树结构变复杂，但符号数减少，整体时间稳定在毫秒级 。

1. 本组代码-本组文本

1. 中文
   信源特性

- **熵 H(X)**: **8.0812 bits/symbol (恒定不变)**

性能对比表

Q	η	原因分析
2	99.71%	二进制霍夫曼接近理论最优
3	98.98%	效率略降，因合并节点时需补虚拟节点
4	98.87%	同上，效率进一步微降
5	89.36%	显著下降，虚拟节点导致冗余增大

2. 英文
   信源特性

- **熵 H(X)**: **4.4790 bits/symbol (恒定不变)**

性能对比表

进制(Q)	平均码长(L)	编码效率(η)	平均比特长度	解码时间(秒)
2	4.5020	99.49%	4.5020	0.0008
3	3.2396	87.23%	≈5.136	0.0006
4	2.5207	88.85%	5.0414	0.0005
5	2.5796	74.78%	≈5.990	0.0004

2. 多组代码-本组文本

指标	组5 (中文)	组5 (英文)	8 (中文)	8 (英文)
信源熵 H(X)	5.0987 bits	2.9100 bits	8.0812 bits	4.6123 bits
平均码长	5.1511 bits	2.9626 bits	8.1043 bits	4.6480 bits

指标	组5 (中文)	组5 (英文)	8 (中文)	8 (英文)
编码效率 (η)	98.98%	98.23%	99.71%*	99.23%*
压缩率 (空间节省)	-	-	65.79%	42.51%
解码耗时	-	-	1.2157 ms	1.6386 ms

> *注：8组的编码效率通过计算得出：
> 中文：η = 8.0812/8.1043 ≈ 99.71%
> 英文：η = 4.6123/4.6480 ≈ 99.23%

五、组间 PK 反馈

（一）对手代码测试情况

交换测试信源后，对手的 Shannon、Huffman 编码代码均可运行，解码准确率 100% 。但在编码效率上有差异，部分对手 Shannon 编码平均码长略长于我方，Huffman 编码效率接近，说明不同实现对概率分布处理、编码树构建细节有别，影响最终性能 。

1. 算法效率对比

算法类型	本组最优效率	对手组最优效率	优势差距
Huffman编码	99.71% (中文)	99.71% (英文)	相同
Shannon编码	97.82% (中文)	96.18% (中文)	+1.64%

> **核心发现**：本组在编码算法上保持效率优势

2. 解码性能对比

性能指标	本组表现	对手组表现
中文解码速度	0.6ms	1.2157ms
英文解码速度	0.8ms	1.6386 ms

> 本组中文解码速度优势最突出，体现核心算法优化深度

（二）优势与不足分析

1. 优势

    1. **解码准确率 100%，算法稳定可靠**
    所有测试（Shannon、Huffman、Q-ary Huffman）中，解码准确率均为 100%，说明编码与解码逻辑实现无误，鲁棒性强。
    2. **Shannon 编码效率略高于对手**
    中文文本效率达 **97.82%**，相比组5的 **96.18%** 提高 **1.64%**，说明在概率计算、码长分配上优化得更好。
    3. **Huffman 编码效率保持理论最优**
    中文效率高达 **99.71%**，与组8相同，证明本组掌握了 Huffman 构造树的关键优化，具备顶级实现质量。

4. **解码速度快，性能优异**
   中文平均解码时间 **0.6ms**，英文为 **0.8ms**，明显优于组8（中文 **1.2ms**，英文 **1.6ms**）。这体现出更高效的解码流程实现，可能包括更优的数据结构（如查找表、字典优化等）。
5. **Q-ary Huffman 多进制支持完整、实验全面**
   涵盖 Q=2,3,4,5，详细比较了不同进制下效率变化和误差来源，说明实现细致且分析严谨。

2. 不足

1. **Q-ary Huffman 英文测试的效率下降较多**
   当 Q=5 时英文编码效率为 **74.78%**，远低于 Q=2 的 **99.49%**，说明在高 Q 下处理稀疏概率分布的能力还有提升空间。

六、总结

本实验验证了 Shannon、Huffman 及 Q - ary Huffman 编码算法的可行性，Huffman 编码因最优性效率更高，Shannon 编码理论基础重要，Q - ary Huffman 适配多进制场景。