一

通过代码对源文本进行处理，可得出**热力图**：

	频数	频率（%）
a	4535	8.47
b	716	1.34
c	1173	2.19
d	2425	4.53
e	5347	9.99
f	434	0.81
g	3643	6.81
h	4347	8.12
i	7377	13.78
j	1886	3.52
k	170	0.32
l	815	1.52
m	743	1.39
n	5742	10.73
o	2876	5.37
p	131	0.24
q	358	0.67
r	423	0.79
s	1565	2.92
t	620	1.16
u	3526	6.59
v	25	0.05
w	335	0.63
x	685	1.28
y	1495	2.79
z	2123	3.97
累计	53515	100

二

定义：

1 均衡性：各个字母的频数标准差，即各个字母的频数越相近则输入越均衡，与频数大小无关；

2 输入效率：输入字母数量/输入汉字数量，即输入效率越高，输入相同汉字时，输入的字母数量越少。

假设：

1) 使用者可以迅速掌握并使用任意合理的正确的编码方案，即任意合理正确的编码方案的效率均是对于计算机运行速度而言的，与使用计算机的人个人习惯等原因无关。
2) 编码方案仅针对源文本进行，是对于源文本较为优秀的编码方案，对于其他文本可能有未编码情况出现。

结论：

据统计，源文档中有：**有 1203 种汉字，**

使用汉语拼音编码为 **53515 个字母，效率为 44.4846，均衡度为 9911.163661**

评价：

原汉语拼音编码方案编码总长度较长，效率较为底下。

三

编码方案：对传统的二叉哈夫曼树进行推广到 **26 叉哈夫曼树**，并针对原文档出现的 **1203 种**汉字的频率进行编码。

输入效率：

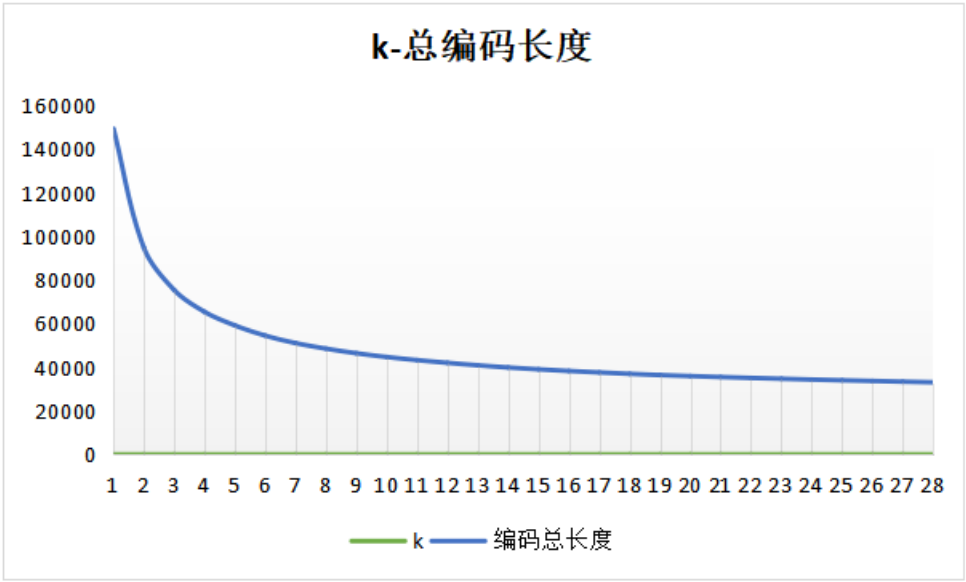
使用 26 叉哈夫曼树，编码长度为 **33989 个字母，效率为 28.2535。**

对比第二问源编码方案：**53515 个字母，效率为 44.4846。**

可见，新的压缩方案有较好压缩效果，更好的输入效率。

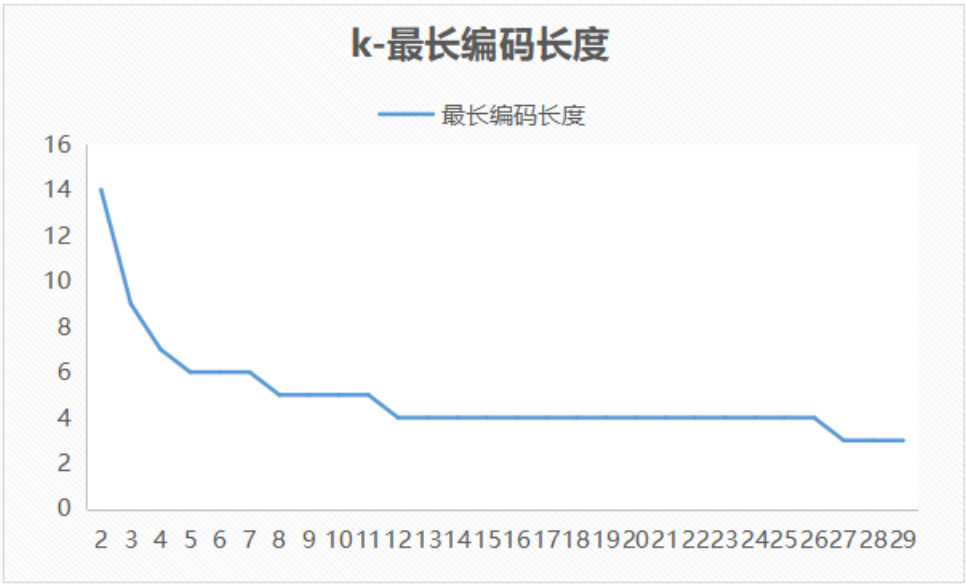
更一般的：

- 测验了 **2-29 叉树**的编码后的输入效率：



可以看出，在 **2-6 叉哈夫曼树**编码时，对于编码输入效率有着较好的改善，此后再逐步增加树枝数时，改善效率**不显著**。

- 每一种编码方式中最长元素编码的长度：



可以看出，在 **2-11 叉哈夫曼树**进行编码时，随着树枝的不断增加，元素最长的编码长度会对应下降，而对于 **12-26 叉树**来说，元素编码长度最长总为 **4 个字母。**

对于均衡性：

使用汉语拼音编码为 **53515 个字母，效率为 44.4846，均衡度为 9911. 163661**

对于 26 叉哈夫曼树编码方案，**由于个人代码能力限制，不足以求出具体数值。**

原因：由于需要给出 26 叉哈夫曼树对于源文本 1203 种汉字的所有具体的编码方案，在定义节点时，需要定义 26 个叶子节点，并在遍历时，需要依次遍历所有节点，并对在查阅相关文献后，个人能力不足以手动写出相应代码。