华东师范大学数据科学与工程学院上机实践报告

课程名称：算法设计与分析	**年级**：22 级	**上机实践成绩**：
指导教师：金澈清	**姓名**：石季凡	
上机实践名称：贪心算法 2	**学号**：10225501403	**上机实践日期**：2023.5.23
上机实践编号：No.11	**组号**：1-403	

一、目的

1．熟悉贪心算法设计的基本思想
2．掌握计算哈夫曼编码的方法

二、内容与设计思想

1. 基于朴素固定长度编码编写字符串编码的代码。
2. 基于贪心算法编写计算哈夫曼编码的代码。
3. 请在网上随意找一些字符串（如文章报道等），对比两种实现方式编码结果长度，计算其压缩比。
4. 对比两种实现方式编码以及译码速度的差异，并简单描述你是如何实现编码和译码的。

三、使用环境

C/C++集成编译环境。

四、实验过程

1. *写出计算两种编码方式的代码。*
2. *用合适的统计图描述你的实验结果。*

五、总结

【各个程序实现思路】：

【统计函数】：对于输入的字符串进行字符分类以及统计，我选择了用链表进行统计，对于字符串 string，每一步取其一个 x 字符，在链表中搜索，如果已经存在则其 frequency 值加一，否则则新建节点，另 w 其 key 为该字符，frequency 为 1，尾插入链表，最终返回链表头 head。

【输出函数】：输出函数共有两个，一个为输出编码表，另一个为输出编码结果。首先输出编码表，输入为一个结构指针数组 tree_1，分别为每一个字符的结构体指针，由于已经编码完成，此时的每个结构体在整体上构成了一棵二叉树，我们只需要对所给的结构体指针数组遍历，每一步操作为向上传递 parent 节点，同时记录每一个传递到的节点的 flag 值，直到传递至 NULL，反向输出我们记录的 flag 值即可，同时把这个编码存入一个组字符串指针数组 codeTab。然后为输出编码结果，通过上述的输出编码表操作，我们得到了一个字符

串指针数组 codeTab，其存放着所有的字符对应的编码，其位置与结构体指针数组一一对应。对于要编码的字符串，我们每读取一个字符，就在结构体指针数组中寻找其位置，再将其位置对应到 codeTab 数组中，输出对应位置的 01 字符串即可。

【朴素等长度编码】：输入字符串 string，首先通过上述的的统计函数得到 string 的字符频率链表，将其分别存入结构体 node 中，node 结构体包括 left 指针、right 指针、frequency、parent 指针、flag 即 01 值，再将结构体对应的结构体指针存入 tree 数组，同时备份一个 tree_1 结构体指针数组。接下来开始构造编码树。我们对 tree 数组进行操作，首先我们定义一个 count 等于数组中结构体指针的个数，定义 int i=0，j=0，然后开始循环，首先每一步循环的第一步为检测 count 值，如果 count==1 则 break，如果 count 为奇数，则令 tree[count++]=NULL，然后开始内循环，内循环的循环条件为 i<count，初始化操作为 i=0，j=0。首先定义一个新结构体指针 newnode，接下来每一次取两个指针，即 tree[i++]作为 newnode 的 left，tree[i++]作为 newnode 的右子，然后初始化 newnode 的其他变量，同时令 newnode 的 left 的 flag 值为 0，right 的 flag 值为 1，将 newnode 放入 tree[j++]。这样一个内循环后，我们将原来的每一个指针两两合并成了一个小子树。随着循环的不断进行，所有的指针最终会合成一个二叉树，每一个叶节点为带有字符的结构体指针或者是 NULL。此时 tree[0]为树的根 root，调用输出函数即可，输入 root 与 tree_1。

【不等长编码】：输入字符串 string，根据上述统计函数获得 string 字符串的字符频率链表，将其分别存入结构体 node 中，node 结构体包括左子指针 left，右子指针 right，父母指针 parent，频率 frequency，01 标识 flag，是否被为空节点 ifempty（用于比较频率相同的情况，即如果一个非叶节点的频率与一个叶节点的频率相同，非叶节点视为更小），存放的字符 key。接下来我们把所有的 node 节点存入 tree 结构体指针数组，并且同时在 tree_1 结构体指针数组中备份一份，此时我们将 tree 数组根据每个结构体的 frequency 来建最小堆，然后进行堆排序（此时是根据 frequency 从大到小排列），接下来每次从中末尾取两个结构体，一个作为新结构体的左子，另一个作为右子，保证左小右大，左子 flag 为 0，右子 flag 为 1，同时新节点的 frequency 等于左右子 frequency 之和，左右子的 parent 为新节点，新节点的 ifempty 为 1，表示其为非叶节点，最后再将该新节点的指针插入 tree 结构体指针数组，并且进行最小堆调整并再次堆排序，重复上述操作，直到 tree 数组中只剩下一个结构体指针，这个指针即为我们哈夫曼编码树的根节点 root，最后调用输出函数即可，输入 root 与 tree_1。

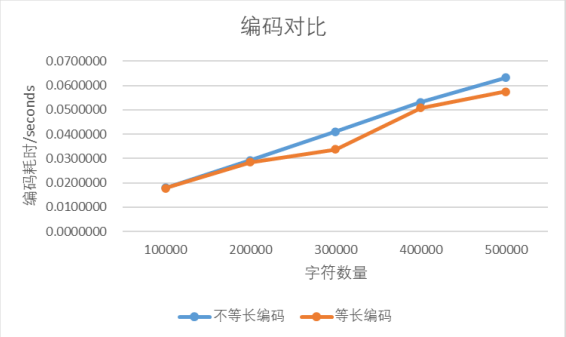
【译码】：输入 k 个字符以及对应的编码，然后输入需要翻译的 01 字符串。我们首先定义一个字符数组 str，初始化每一个位置为‘\0’，对于一个字符 ch 以及其对应的编码，我们定义一个 locate=0，逐一读取其编码，若读到 0 则 locate=loca*2+1，为 1 则 locate=locate*2+2，直到读完后把 ch 存入 str[locate]，将 locate 初始化为 0，重复上述操作，直到所有字符都被存入 str 数组。接下来读取需要翻译的 01 字符串，定义一个 locate=0，同样的，我们读到 0 则令 locate=locate*2+1，读到 1 则令 locate=locate*2+2，每读一个字符检查 str[locate]是否有存入的字符，若有则输出该字符，同时令 locate=0，不断循环直到读完整个输入的 01 字符串，最终的输出结果即为我们需要的翻译出来的结果。

【实验】：

【编码时间对比】：分别输入长度为 1e5,2e5,3e5,4e5,5e5 个字符的字符串进行压缩，并计时对比。

得到结果如下，其中耗时的单位为秒，字符来源为网络以及本人英语作文的拼凑。

	不等长编码	等长编码
100000	0.0178989	0.0178118
200000	0.0291887	0.0285161
300000	0.0410575	0.0337991
400000	0.0530711	0.0508062
500000	0.0632028	0.0574857



【压缩率对比】：我从 China Daily 中复制了一段文字，共计 2310 个字符，得到结果如下

```
His remarks strongly resonated with the leaders of Central Asian countries. During the summit, they voiced firm support for each
other to choose a development path compatible with their national conditions, firmly uphold their core interests including
sovereignty, independence, security and territorial integrity, and oppose interference in others' internal affairs.Central Asian
countries fully recognize the significance of the Chinese path to modernization for the world's development, and reiterate their
firm commitment to the one-China principle. As forces for peace and justice, China and Central Asian countries, with mutual
understanding and joint efforts, will surely inject more positive energy and stability into this complicated and volatile
world.During the summit, China and five Central Asian countries signed seven bilateral and multilateral documents as well as over
100 cooperation agreements in various fields. The achievements and impacts of the summit were unprecedented.The summit established
a comprehensive framework for the mechanism. Taking the opportunity of this summit, the six countries have officially inaugurated
the China-Central Asia Summit Mechanism, with China and Central Asian countries taking turns to host the biennial summit.Head-of-
state diplomacy will keep offering strategic guidance on enhancing top-level planning and coordination for China-Central Asia
relations.The first China-Central Asia Summit was a grand meeting that blended history and the future. Xi'an, as the starting point
of the ancient Silk Road, has once again become a starting point in the new era witnessing the joint efforts of China and Central
Asian countries to create a brighter future.The summit demonstrated unity, creativity and efficiency, creating a new platform and
opening up new prospects for China-Central Asia cooperation. It marked a new milestone in the development of China-Central Asia
relations.It is believed that with joint efforts, China's relations with Central Asian countries will forge ahead like a ship
braving all winds and waves, offer new vitality to the development and revitalization of the six countries, inject strong, positive
energy to peace and stability of the region, and make new contributions to building a community with a shared future for mankind.
```

```
Haffman finished!
time_cost= 0.0005324 second, length_average = 4.35773062

请按任意键继续. . .
```

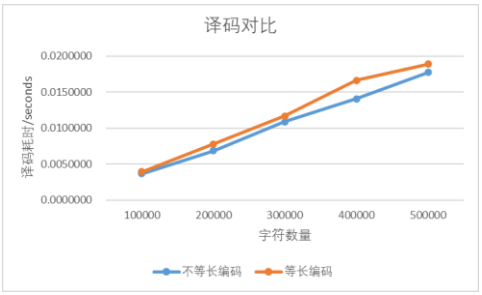
```
Normal finished!
time_cost= 0.0005038 second, length_average = 6.00000000

请按任意键继续. . .
```

计算得到压缩比为 Haffman/normal=0.72628844，由此可见哈夫曼编码确实相较于等长度编码压缩效果更好。

【译码时间对比】：我们将编码实验的结果进行译码，并计时，可以得到如下结果，其中耗时的单位为秒。

	不等长编码	等长编码
100000	0.0036672	0.0039339
200000	0.0068483	0.0077699
300000	0.0108559	0.0116833
400000	0.0140737	0.0166382
500000	0.0177562	0.0188673



【实验结果总结】：根据上述实验结果我们可以发现，在编码时，哈夫曼编码要慢于等长编码，这是由于哈夫曼编码的节点合并需要不断地进行堆排序堆调整，因此耗时较长，而等长度编码仅需要合并即可。同时由于哈夫曼编码对于字符编码根据出现频率地合理分配，即频率高的短，频率低的长，使其压缩的效果更好。在译码中，由于原理而言，二者在每一步的操作上毫无区别，但是由于哈夫曼编码的结果更短，因此译码耗时更短。总而言之，哈夫曼编码是一个时间换空间的策略，利用更为复杂的调整去换取空间上更少的消耗。