

**数据压缩作业1报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | 算术编码的实现 | | | | | |
| 姓名 | 石文杰 | | 院系 | 计算学部软件工程系 | | |
| 班级 | 1837102 | | 学号 | 1183710218 | | |
| 任课教师 | 刘岩 | | 指导教师 |  | | |
| 实验地点 |  | | 实验时间 |  | | |
| 实验课表现 | 出勤、表现得分(10) |  | 实验报告  得分(40) |  | 实验总分 |  |
| 操作结果得分(50) |  |
| 教师评语 | | | | | | |
|  | | | | | | |

|  |
| --- |
| 实验目的： |
| 理解算术编码的基本原理，体会数据压缩的实际过程。 |
| 实验内容： |
| 实验要求：输入一个长度为n的字符串，使用算术编码方式将其压缩为一个小数code，再将其解码为原字符串。  算术编码的原理：  算术编码是一种无损压缩的编码方式，流程是这样的：首先计算字符串中各个字符出现的概率，形成一个概率分布表，随后取出字符串中的一个字符，得到它所在的区间，并和当前的区间（初始为[0,1））做一个投射，即将字符所在区间与当前区间做合成，然后取出下一个字符，重复这一流程，直到字符串结束，得到了最终的区间，则这个区间内的任意一个小数都能表示这个字符串的所有字符信息，所有的信息都得到了保留，所以算术编码是无损压缩的编码方式。  在每次取出字符，做区间合成时，既保留了之前的字符顺序信息，又将当前的字符信息以其用概率区间缩小当前区间范围的方式也保留在了新的当前区间中，没过一次迭代，小数区间就能多表示一条字符信息  解码时，找到code所在的区间，即可得到对应于区间的一个字符，将code从这个区间中剥离出来得到新的区间，再得到新的字符，持续这个循环，直到字符串达到输入的长度 |
| 实验过程： |
| 编码：   1. 首先遍历一遍字符串，得到各个字符的出现的频率 2. 遍历字符频率表，使用频率除以长度的方式计算每个字符的概率，生成区间分布表 3. 进行编码，使用公式：   low = low + (high - low) \* L  high = low + (high - low) \*H  进行迭代，其中，low和high分别是当前区间的下界和上界，L和H分别是当前字符所在区间的下界和上界  解码：   1. 创建一个空串 2. 查找区间分布表，循环地判断参数code属于哪个区间，得到区间对应的字符，将其插入空串后面 3. 使用公式code = (code – L) / (H - L)将新的区间从原区间剥离出来 4. 如果当前字符串没有达到原字符串长度，则回到第二步   源代码及注释请参见本文件夹下的ArithmeticCoding.java |
| 实验结果： |
| 采用演示截图、文字说明等方式，给出本次实验的实验结果。    如图所示，程序输入一个长度为250的无规律字符串作为测试，解码之后得到的decode\_string与原始字符串source\_string完全相同，程序成功完成了编解码。 |
| 问题讨论： |
| 对实验过程中的思考问题进行讨论或回答。   1. double精度受限，如果使用double类型作为每个字符概率及区间表示，最多只能保证8到9位的字符解码正确，如何解决？   我的解决办法是使用Java中的精确浮点表示类型BigDemical，成功解决了这个问题，在数百位字符的测试用例中均正确无误，但是由于BigDemical内部方法复杂的实现机制，导致编解码效率的降低，如果要减少这种效率的损失，可能要自己实现一个更针对于算术编码的特性功能更单一的类   1. 计算每个字符出现的概率时很容易产生循环小数，如何决定保留的概率精度？   最开始，我尝试使用当前待编码字符串长度作为精度保留，后来发现极为不妥，编码产生的小数十分之长，导致压缩完全没有意义，反而产生了一个更复杂的信息序列。  经过反复思考，我认为，对于一个长度为N的字符串，小数的精度保留为[log10（N）]为宜，原因在于：设小数精度为k，那么这个小数大致能表示10的k次幂长度字符串中每个字符的概率精确，所以对于数百字符长度的字符串，我选择使用的小数精度保留为3，这样能保证1000字符长度以内的字符串能被正确编解码，如果超过这个精度，尽量不要设置太过，例如对1000字符长度之内的字符串设置精度为5或更高，则大大浪费了计算资源，也极大地减少了压缩的效率，甚至可能发生压缩后数据长度更长的情况。   1. 能否将出现过的字符概率直接设置为均等，因为区间实际上才是真正对应唯一的字符，计算每个字符的概率的意义是什么   我觉得，将所有字符的概率设置为均等实际上是可行的，但是计算每个字符的概率依然有其原因  对于出现频率高的字符，设置更大的区间，能减少编码时区间缩小的速度，那么计算所有字符的概率，并将概率作为区间大小，则保证了最终编码字符串完成后得到的小数区间最大化，使得我们编码得到的小数可以有最小的位数，等于是减少了冗余的信息量，提高了压缩的效率。 |
| 心得体会： |
| 结合实验过程和结果给出实验的体会和收获。  对于算术编码的实现思路并不复杂，但是很多细节之处仍然值得仔细探究，我觉得从中体会到了将数学与编程相结合的一些乐趣，对于数据压缩有了更深更实际的理解。 |