数据存储是以字节为单位，一个字节占八个比特位（即八个二进制数）每个字符（详见ASCII码表）都有对应的二进制编码，而我们通过哈夫曼编码给每个字符相对应的编码，以这样的编码存储在计算机里就会节省了很多。

比如字符a在计算机以10100000存储，而假设我们通过哈夫曼编码给a编码为01，以01的形式存储在计算机就会节省空间，数据越多效果越明显。

主要的算法：哈夫曼建树，编码，解码

哈夫曼建树：

首先要有节点数，节点权值。主要思想：权值越大的靠前，小的靠后，所以每次找两个权值最小的建树，

而对于字符来说，存储在计算机里的是对应的ASCII码值对应的二进制数，所以当给一段文字存储在计算机中，每个字符都要占一个或者两个字节（汉字占两个字节）。但通过建哈夫曼树，给每个字符编码（我这里是给字节为编码的，因为汉字占两个字节不好统计他的权值，而用字节的话，就直接把它作为辅助结构体找的下标，当扫描时，下标一样时即ASCII码值、对应字符一样时加一，这样一次扫描后就能统计各个字符的权值了），