Huffman Code編碼範例TEXT：adbeedbaegfgabeacdha

步驟1：列出所有字元(word)出現的次數，並記錄到array x[8]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x[0] ”a” | x[1] ”b” | x[2] ”c” | x[3] ”d” | x[4] ”e” | x[5] ”f” | x[6] ”g” | x[7] ”h” | total |
| 5 | 3 | 1 | 3 | 4 | 1 | 2 | 1 | 20 |

步驟2：因為有8種字元，每一次合併一個，共需合併 (8-1)=7次。產生array temp[7][2]，每次從x[] 中找出數值最小的兩個相加合併到兩個中index較小者，空的設為 total+1，並將合併的兩個index 存入temp[][0] 與 temp[][1]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x[index] | n=0 | n=1 | n=2 | n=3 | n=4 | n=5 | n=6 | n=7 |
| x[0] “a” | 5 | 5 | 5 | 5 | 5 | 5 | 11 | 20 |
| x[1] “b” | 3 | 3 | 3 | 3 | 6 | 6 | 21 | 21 |
| x[2] “c” | 1 | 1 | 3 | 3 | 21 | 21 | 21 | 21 |
| x[3] “d” | 3 | 3 | 3 | 5 | 5 | 9 | 9 | 21 |
| x[4] “e” | 4 | 4 | 4 | 4 | 4 | 21 | 21 | 21 |
| x[5] “f” | 1 | 2 | 2 | 21 | 21 | 21 | 21 | 21 |
| x[6] “g” | 2 | 2 | 21 | 21 | 21 | 21 | 21 | 21 |
| x[7] “h” | 1 | 21 | 21 | 21 | 21 | 21 | 21 | 21 |
| temp[n][0] | 5 | 2 | 3 | 1 | 3 | 0 | 0 | -- |
| temp[n][1] | 7 | 6 | 5 | 2 | 4 | 1 | 3 | -- |

最後得到 temp[][]，其中temp[][]，最長的碼可能為 =5<8

步驟3：從 temp[6][] 反推到 temp[5][] 最後到temp [0][] 填入碼(code)到array c[8][5]，每次給予temp[][0]者0，temp[][1]者1。因為合併皆合併到index較小者，所以遇到之前已經有給予值的index(已經有值的c[index])，需繼承index較小者的碼 (依照上述作法不可能會出現兩者皆已有值的情況)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| temp[n][0] | 0 | 0 | 3 | 1 | 3 | 2 | 5 |  |
| temp[n][1] | 3 | 1 | 4 | 2 | 5 | 6 | 7 |  |
| c[index] | n=6 | n=5 | n=4 | n=3 | n=2 | n=1 | n=0 | c[][] |
| c[0][] “a” | 0 | 00 |  |  |  |  |  | 00 |
| c[1][] “b” |  | 01 |  | 010 |  |  |  | 010 |
| c[2][] “c” |  |  |  | 011 |  | 0110 |  | 0110 |
| c[3][] “d” | 1 |  | 10 |  | 100 |  |  | 100 |
| c[4][] “e” |  |  | 11 |  |  |  |  | 11 |
| c[5][] “f” |  |  |  |  | 101 |  | 1010 | 1010 |
| c[6][] “g” |  | 1 |  |  |  | 0111 |  | 0111 |
| c[7][] “h” |  |  |  |  |  |  | 1011 | 1011 |

最後得到編碼c[][]。

* 實作時需注意文字檔中的空格，標點符號與大小寫。
* 可以利用二元樹(binary tree)的方式儲存編碼(可參考wiki-- Huffman code)。
* 解碼可利用查表或二元樹查詢