Huffman Algorithm壓縮程式

|  |  |
| --- | --- |
| 開發環境 | |
|  | Microsoft Windows 7專業版(Professional) |
|  | Microsoft .NET Framework, Version 4.6.01055 |
|  | Microsoft Visual Studio Community 2015, Version 14.0.25431.01 Update 3 |

|  |  |
| --- | --- |
| 使用語言 | |
|  | Microsoft Visual C# |

|  |
| --- |
| 簡介 |
| Huffman Algorithm是一種無失真資料壓縮編碼的演算法，其原理是將檔案中出現頻率較高的字元使用較短的編碼，而出現頻率較低的字元使用較長的編碼。利用此種動態長度編碼方式取代每個字元固定長度的編碼，來降低整體檔案平均的編碼長度達到無失真壓縮的目的。  要取得每個字元的Huffman Code必須先計算每個字元的出現頻率來建立Huffman Tree，而建立Huffman Tree的過程中必須使用Min Heap的資料結構來進行排序與插入移除，如此令Huffman Tree的建構能在O(n logn)的時間複雜度完成。  由於不同的檔案經過Huffman Algorithm會產出不同的Huffman Code，因此壓縮檔內必須存放Huffman Code的解碼表才能正確的進行解壓縮，導致壓縮檔勢必因為存放解碼表而增加額外的容量。所以已經經過壓縮處理的檔案格式如rar、jpg等，再使用Huffman Algorithm壓縮時所產生的壓縮率非常有限，甚至因解碼表大小超過壓縮率節省的空間，造成壓縮後的大小比原始檔案來的大的現象。  為了減少解碼表所佔用的空間，本程式實作的編碼方式為Canonical Huffman Code，此種特殊的編碼是透過對原本的Huffman Code進行些微調整就可得到，可以有效的減少存放解碼表的空間。 |

|  |  |
| --- | --- |
| 測試檔案 | |
|  | 本作品額外提供三個輸入檔案作為壓縮測試檔之用： |
|  | ‧邱經達resume.docx  ‧邱經達resume.pdf  ‧邱經達resume.bmp |

|  |  |
| --- | --- |
| 程式介面與執行結果 | |
|  | 說明 |
|  |  |
|  | 壓縮 |
|  |  |
|  | 解壓縮 |
|  |  |