**Algorithms**

Programming Assignment #2 Report

Ambitious DVD Dealer

**B02901135 電機三 陳奐晴**

**Jobs:**

1. **Greedy:**

**STEP 1**

我使用的greedy方法是先開一個歌曲數量大小(假設有n首)的array，將歌曲複製進去之後依照歌曲長度用insertion sort 將歌曲sort好。

**STEP 2**

接下來使用binary search，先猜測可以放進一半數量的歌曲數(n/2首)，找出不超過最中間那首歌(第n/2首歌)大小的所有歌曲，依照歌曲原先順序選擇放入CD內。如果放得進有可能所猜測之n/2太小，則向右再切一半猜測(猜測可放入3n/4首)，如果放不盡表示所猜測之n/2太大，則向左切一半猜測(猜測可放入n/4首)…依此類推最終找到解。

**Complexity**

1. Copy the songs into new array ---- Θ( n )
2. Insertion Sort ---- O( nlgn)
3. Binary Search ---- O( lgn
4. 按照順序將歌曲放入CD ---- O( n )

🡪Overall: O( nlgn )

1. **Mathematical Analysis:**

**Prove:**

**Optimal substructure:**

假設Ｆ是總共Ｎ首歌Ｍ張DVD最多能含有的歌曲數，那必定有F’首歌是總共Ｎ首歌Ｍ-1張DVD最多能含有的歌曲數。

**Proof cut and paste:**

若Ｆ’不是Ｎ首歌Ｍ-1張DVD最多能含有的歌曲數，則用F’’替換。當再多加一張DVD變成M張，Ｆ就是最佳解，與先前假設矛盾，故得證。

**Recursive:**

1. **Complexity:**

因為是3 dimension，complexity為O(N\*M\*T)

測資數據結果:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| case | GD | | | DP | | |
|  | Maximum songs | CPU time(s) | Memory(KB) | Maximum songs | CPU time(s) | Memory(KB) |
| case1  N= | 1 | 0 | 13.284 | 1 | 0 | 13.284 |
| case2  N= | 0 | 0 | 13.284 | 0 | 0 | 13.284 |
| case3  N= | 0 | 0 | 13.284 | 0 | 0 | 13.284 |
| case4  N= | 263 | 0 | 13.284 | 285 | 0 | 18.432 |
| case5  N= | 10012 | 1.484 | 14.456 | 10265 | 25.6351 | 14407.3 |
| case6  N= | 12023 | 1.468 | 14.456 | 13102 | 38.7121 | 22693.3 |
| case7  N= | 1000 | 0 | 13.284 | 1000 | 0.168 | 162.004 |
| case8  N= | 0 | 0 | 13.284 | 0 | 0 | 13.684 |
| case9  N= | 61 | 0 | 13.284 | 62 | 0 | 13.676 |

測資數據圖表: