

演算法作業說明

陳鈺鈞 (M113040018@student.nsysu.edu.tw)

林正昊 (M113040042@student.nsysu.edu.tw)

陳宥呈 (M113040112@student.nsysu.edu.tw)



演算法作業1

Exhaustive Search 解 0/1 Knapsack problem

作業1 - Exhaustive Search (1/2)

- 嘗試所有可能的解
- 假設有4項物品 (0 → 不選 1 → 選)

e.g. 可能的解

0000

0001

⋮

1110

1111


作業1 - Exhaustive Search (2/2)

- 計算背包物品總價值注意事項
 - 確認所有物品是否皆能夠裝入背包
 - 若物品總重量 \leq 背包容量限制，則計算其總價值
 - 若物品總重量 $>$ 背包容量限制，則直接將總價值設為0

演算法作業2

Greedy 解 0/1 Knapsack problem

- 每次皆選擇 CP值最高 的放進背包內


$$\frac{value}{weight}$$

- 若選擇到的物品無法放進背包 (重量大於背包剩餘空間)，則選擇CP值次高的，直到所有物品皆無法放進背包

作業繳交規則

- 繳交作業格式
 - 一律壓縮成：學號_hw~~x~~.zip (e.g. b103040000_hw1.zip)
 - 壓縮檔須包含 (請勿包含.exe)
 1. 程式碼 (C or C++)
 2. 輸出檔 (ans.txt)
- 繳交方式：上傳網路大學
- 繳交期限：10/26 (四) 23:59
- 實體demo位址：EC5009-1 (請自行攜帶電腦)
- 實體測驗時間：10/16,10/17,10/19,10/24,10/26,
下午1:30 ~5:00，依公告為主

程式是否能正確執行?	30% (不能執行則全部拿0分)
答案是否正確?	20% (答案錯最多拿50分)
程式撰寫之結構與邏輯是否正確?	20%
輸出結果是否完整?	10%
清楚表達程式流程? (口頭 or 註解)	10%
繳交格式是否正確? (檔案名稱 and 檔案格式)	5%
是否能動態讀入readfile?	5%

※ 所有項目均為部分給分

- 3筆測試資料 (每筆資料物品數量不同)
 - item.txt : 物品重量、價值
 - ans.txt : 最佳解 (僅前兩筆測試資料提供)

10 165

23 92

31 57

29 49

44 68

53 60

38 43

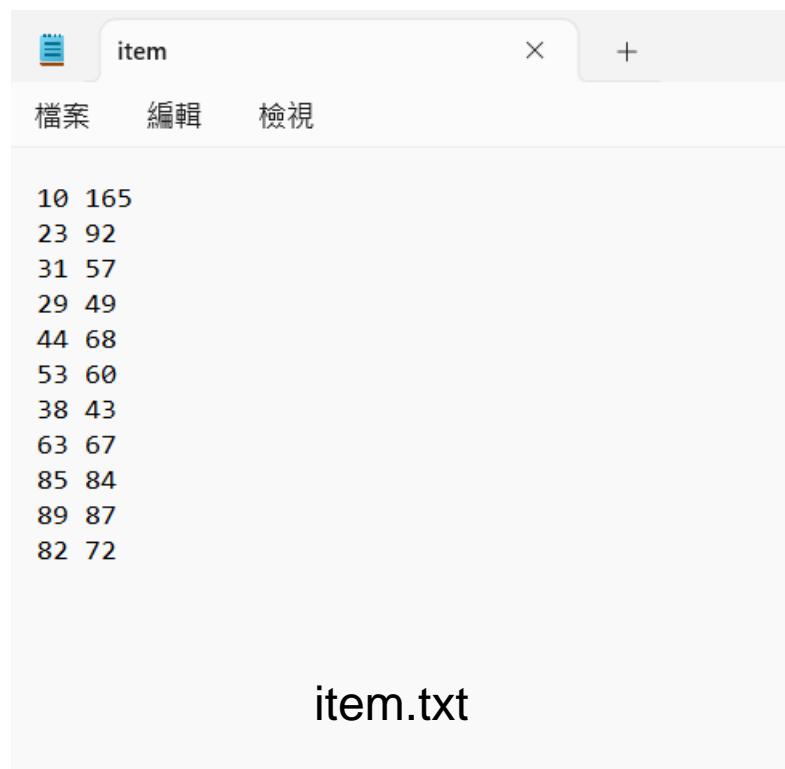
63 67

85 84

89 87

82 72

物品數量 (number of items)	10
背包最大容量 (capacity)	165
物品重量 (weight)	[23,31,29,44,53,38,63,85,89,82]
物品價值 (value)	[92,57,49,68,60,43,67,84,87,72]



- 每筆測試資料皆獨立輸出
 - e.g. ans_dt01.txt, ans_dt02.txt, ans_dt03.txt
- 輸出規定
 1. 輸出背包所裝的所有物品總價值
 2. 以0和1表示物品是否被選擇

輸出範例

- max profit:309
- solution:1111010000



```
max profit:309  
solution:1111010000
```



Thank You ;-)