作業規定

使用語言:

C or C++

欲解決問題:

利用 tabu search 演算法找出 0/1 knapsack problem 之最佳解(或接近最佳解) ，tabu search 之 iteration 次數設定為 1000，tabu search 之 Bits 數量可參考下列測資進行設定。

*tabu search 演算法之作法及定義可於附檔 Essentials-metaheuristic-algorithm.pdf 內找到。

程式規定:

- 作業內請包含原始碼、編譯後執行檔及 readme.txt 文字檔，並打包為一份壓縮檔。readme.txt 檔案內說明程式的執行方式及記錄執行結果，執行結果請紀錄找到最佳的解並且截圖，格式不限。
- 程式碼內請加上註解。

0/1 Knapsack Problem:

Given N items where each item has some weight and profit associated with it and also given a bag with capacity W, [i.e., the bag can hold at most W weight in it]. The task is to put the items into the bag such that the sum of profits associated with them is the maximum possible.

Note: The constraint here is we can either put an item completely into the bag or cannot put it at all [It is not possible to put a part of an item into the bag].

Input Format:

number(N) weight(W)

w_1 p_1

w_2 p_2

w_3 p_3

⋮

w_N p_N

Output Format:

max profit:$maximize \sum_{i=1}^{N} x_i\, p_i, subject\ to \sum_{i=1}^{N} x_i\, w_i \le W, x_i \in \{0,1\}$

solution:$x_1 x_2 x_3 \ldots x_N, x_i \in \{0,1\}$

Example1:

Input data: N = 3, W = 4, profit = {1, 2, 3}, weight = {4, 5, 1}

Input

3 4

4 1

5 2

1 3

Output

max profit:3

solution:001

Example2:

Input data: N = 3, W = 3, profit = {1, 2, 3}, weight = {4, 5, 6}

Input

3 3

4 1

5 2

6 3

Output

max profit:0

solution:000

Test data:

Input data: N = 10, W = 165,

profit = {92,57,49,68,60,43,67,84,87,72}, weight = {23,31,29,44,53,38,63,85,89,82}

Output

max profit:309

solution: 1111010000