FIT1045 Algorithmic Problem Solving – Workshop 6.

Objectives

The **objectives of this workshop** are:

- To get familiar with greedy strategy.
- To implement greedy algorithms to solve problems.
- To develop skills and understanding of practical uses of greedy approach.

Useful Material

Greedy algorithm: `https://en.wikipedia.org/wiki/Greedy_algorithm`

Task 1:

A thief is robbing a store and can carry a maximal weight of W into his knapsack. There are n items available in the store and weight of ith item is w_i and its values is v_i. The thief cannot take a fraction of any item, that is either the thief has to take the whole of it or nothing. The items descriptions are given in the file items.txt, one item on each line. What items should the thief take, if the thief decides to take a greedy aprroach? (Note that the exact kind of greed is not specified. There are some examples of different possible greedy approaches given in lectures)

To solve this problem,

- Write a function `greedy_items(values, weights, capacity, pref)` which takes as input a list of the weights of the items and a list of their values, where `weights[i]` is the weight of item `i` and `values[i]` is the value of item `i`. The function should return a list of the indices of the items the thief should take (similar code is given in lectures).
- Write a function `process_file(filename)` which takes as input a string which is the name of a file. `process_file` should process and read the contents of that file into some appropriate data structures (that it returns) which can then be given to `greedy_items` in order to solve the problem specified in the file.
- Write some code which reads the file name and capacity from the user, and then runs your functions to solve the problem. Your code should display the result to the user as shown below.
- (Extension, not ncesssary for marks) Modify the `greedy_items` so that it takes a fourth arguement, `pref`, as shown in lectures. See if you can come up with appropriate `pref` arguements so that your greedy function works on least weight, most value, inverse weight, and most value/weight.

Sample output:

```
Please enter file name with item details: items.txt
Please enter the capacity of the knapsack: 7 Kg
Optimal answer: item 2 and item 3, value =30$
```

Task 2:

Part 1: Given some coin denominations, your goal is to make change for an amount of money using the smallest number of coins. Write a function `greedy_coin_change(amount, denoms)` which takes as input an amount of money and a list of numbers representing the various coin denominations you can use. The function returns a list of numbers which represent how many of each coin are needed to make up the given `amount`. The function should use a greedy algorithm to solve this problem. **You may assume that `denoms` is sorted in ascending order when it is given to your function.**

Example: Calling `greedy_coin_change(15, [1, 7, 13])` would return `[2, 0, 1]` indicating that you need 2 coins of value 1, no coins of value 7 and 1 coin of value 13 to make 15.
Note: We covered the coin changing problem in tutorial 5, feel free to refer back to that as it may help you here