Instituto Politécnico Nacional
Escuela Superior de Cómputo
Evolutionary Computing Python & Greedy Algorithm
Alumno: Rodríguez Coronado Ricardo Profesor: Jorge Luis Rosas Trigueros
Realización: 04/feb/20 Entrega: 11/feb/20

Theoretical framework

Python

Who created python?

- Guido Van Rossoum, Netherlands, 1990

Explain the game of the name

- Python was named in honor to the comedy group Monty Python

What is the current version of python?

- 3.8.1

Explain the term "pythonic"

- It is a philosophy of the programming language, such as:
  - Beautiful is better than ugly.
  - Explicit is better than implicit.
  - Simple is better than complex.
  - Complex is better than complicated.
  - Readability counts.

Explain the difference between the following

- List: can contain mixed types and are mutable
- Tuple: Can contain mixed types, but are immutable
- Set: Unordered set, contains no duplicates; can contain mixed types, if hashable and immutable
- Dictionary: Associative array (or dictionary) of key and value pairs; can contain mixed types (keys and values), keys must be a hashable type, mutable
- Array: The main difference between a list and an array is the functions that you can perform to them, it’s usually homogeneous

Knapsack problem

Given a ser of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less or than or equal to a given limit and the total value is as large as possible.

In the 0/1 KP the number of each item can only be 0 or 1

Changing-making problem

Also know as minimum coin change problem, addresses the question of finding the minimum number of coins (of certain denominations) that add up to a given amount of money

Greedy Algorithm

Algorithm paradigm, that follows the problem solving idea of making the locally optimal choice at each stage with the intent of finding a global optimum, for many problems a greedy strategy may not produce an optimal solution, but nonetheless, a greedy approach may yield locally optimal solutions in a reasonable amount of time.

Material and equipment:
Python 3.0+
A text editor
PC or Laptop with: Windows 7+/Linux/MacOS X

Practice development
Write a greedy algorithm in python for

- 0/1 Knapsack problem

  To start, we have two variables in the problem, the maximum capability of a knapsack and a list of objects, which have a weight and a value; we need to maximize this, so each object will be represented as a duple
  The first step is sorting the objects, by value, in descending order.
  Then we need to declare three variables, one will keep the objects that are in the knapsack; 'mochila', other that will keep the weight in each iteration, and one last; 'valor', will keep the value maximum.
  The next step is to use a 'for' cycle, in which, each iteration will handle an object; 2-tuple, different. Nested to this, a conditional is used, responsible for verifying that the weight in the backpack added to the weight of the object being iterated, is less than or equal to the backpack capability, if so, the 2-tuple is added to the variable 'mochila', the weight of the object is added to the variable 'peso' and finally, the value is added to the variable 'valor'.

- Changing-Making Problem

  In this case, two input data are handled, a list of coin denominations and the total change.
  Like the previous problem, two variables are necessary, one called 'denominaciones', which will have all the currencies that will be given for exchange and 'cambio' that will store the amount of changing with each iteration.
  We begin by ordering the list of denominations in descending order, after a 'for' cycle is necessary, in which each iteration will be a denomination. Nested to this, a 'while' cycle, whose entry condition is that the sum of the 'cambio'

variable and the denomination of the current iteration, be less than or equal to the total change, within this, it is only necessary to add the denomination to the list 'denominaciones' and add the value of the denomination to the variable 'cambio' .

Conclusions

The greedy algorithm can solve different problems quickly, however, it has limits, among which are the fact that only one possible solution is found, but depending of the problem may be optimal or not, another is the fact that in some problems; Changing-Making Problem, for example, it is necessary to have a base case for the resolution of this, since otherwise, the algorithm cannot work.

Bibliography

Python (Programming Language). (2020). In Wikipedia. *Retrieved* 09 February 2020 https://en.wikipedia.org/wiki/Python_(programming_language)