

Costa da Quinta, Joao Filipe

TP1 : Geedy

(1)

coin_set=[30,24,12,6,3,1]

Fonction compute_change (money, coin_Set)

compteur=0

change=[]

while i < len(coin_set)

 v = coin_set[compteur]

 if money > ou = v faire:

 change.append(v)

 money = money - v

 else incrementer:

 i=i+1

else fin while

return change

(2)

On peut dire que cet algorithme est de type greedy car à chaque étape il fait un choix optimal.

À chaque étape, cet algorithme choisit de rendre la plus grande pièce de monnaie possible.

Par exemple, pour rendre une monnaie de [30], il rend une pièce de [30], plutôt que 30 pièces de [1].

(3)

Non, il n'est pas toujours optimal.

Soit money = 48,

L'algorithme va rendre une monnaie de [30,12,6], par contre la réponse plus optimale, c-à-d rendre le moins de pièces possible, serait de rendre [24,24].

Pour cet algorithme la meilleure réponse est de à chaque fois rendre la plus grande pièce possible, en commençant par rendre une pièce de 30, il est ensuite obligé de rendre 12 et 6, plutôt que [24,24]. C'est pour cette raison qu'on peut dire qu'il optimise chaque étape.