

info401 : Programmation fonctionnelle
TD 7 : algorithmes gloutons, programmation dynamique

Pierre Hyvernât
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 22, poste : 94 22
email : Pierre.Hyvernât@univ-savoie.fr
www : <http://www.lama.univ-savoie.fr/~hyvernât/>

La boulangère dispose d'une quantité illimitée de pièces de 1, 2, 5 et 10 centimes. Elle peut obtenir n'importe quelle somme à partir de ces pièces, mais comment faire pour minimiser le nombre total de pièces utilisées ?

Question 1. Proposez un algorithme simple pour trouver une solution optimale. Programmez une fonction correspondante :

```
monnaie_glouton : int list -> int -> (int*int) list
```

devra prendre en argument la liste des pièces qu'on a le droit d'utiliser et le montant total qu'on veut obtenir. Le résultat sera une solution qui minimise le nombre total de pièces utilisées.

Par exemple : `monnaie_glouton [1 ; 2 ; 5 ; 10] 9` donnera `[(5,1) ; (2,2)]` car il suffit d'une pièce de 5 centimes et de 2 pièces de 2 centimes pour faire 9 centimes.

Question 2. Sur la planète kcahteN, les habitants utilisent des pièces de 1, 4 et 6 dimkroz. Faites fonctionner votre fonction sur des petites valeurs (entre 1 et 10). Que constatez-vous ?

Question 3. Pour corriger le problème de la question précédente, nous allons utiliser un algorithme récursif plus complexe. Nous allons commencer par seulement calculer le nombre total de pièces utilisées :

```
compte_monnaie 0 _ = 0  
compte_monnaie _ [] = "+infini"  
compte_monnaie n (p::l) = min (compte_monnaie n l)  
                           (1 + (compte_monnaie (n-p) (p::l)))
```

Le premier cas correspond au cas où on n'utilise pas la pièce p, et le second au cas où on utilise la pièce p.

Écrivez la fonction `monnaie_rec` correspondante.

Question 4. Généralisez ce principe pour obtenir le nombre de chacune des pièces, et écrivez la fonction correspondante :

```
monnaie_rec : int list -> int -> (int*int) list
```

Question 5. Écrivez une version mémoisée de cette fonction.

Question 6. À votre avis, quelle sera la complexité de ces fonctions :

- `monnaie_glouton`,
- `compte_monnaie`,
- `monnaie_memoisee`.