

Le compte est Bon

Par Michel Grolet et Antoine Chevaleyre.

- [Le compte est Bon](#)
 - [Fonctionnement global de l'algorithme](#)
 - [Optimisation du nombre d'itérations](#) - [Comment obtenir moins de couples](#) - [Comment passer certaines opérations](#) - [Enlever l'appel récursif si on n'a plus que 3 nombres](#)
 - [Détail des fonctions](#) - [main](#) - [genererNombres](#) - [calculerCompteEstBon](#) - [calculerCouples](#) - [calculer](#) - [afficherResultat](#)

Fonctionnement global de l'algorithme

La fonction principale de `CompteEstBon.java`, à savoir `calculerCompteEstBon()`, est appelée par la méthode `main`.

La fonction `genererNombres(args)` vérifie que les arguments passés à `CompteEstBon` sont valides. Si c'est le cas, elle met le nombre attendu dans un entier et retourne les 6 nombres de départ sous forme de liste.

Optimisation du nombre d'itérations

Plusieurs ajouts dans `calculerCompteEstBon()`, `calculerCouples()` et `calculer()` ont permis de réduire drastiquement le nombre d'itérations.

Comment obtenir moins de couples

Il faut déjà penser aux deux `for` de `calculerCouples()`. Si on les agence bien et qu'on retire les couples où `a=b`, on tombe à 2 parmi (nombre de valeurs)) couples (soit 15 couples au début, puis 10 couples, ...)

On peut retirer de ces couples ceux où `a` ou `b` sont nuls, car ils n'apportent rien à l'algorithme (les opérations avec un membre nul sont inutiles).

Comment passer certaines opérations

On n'a pas besoin de faire `a*b` et `a/b` si `a` ou `b` sont nuls ou égaux à 1. On doit aussi retirer les divisions `b/a` car la liste est triée dans l'ordre décroissant et donc le résultat ne serait pas un entier.

Enlever l'appel récursif si on n'a plus que 3 nombres

Si on n'a que 3 valeurs, il n'est plus utile de faire un appel récursif.

Détail des fonctions

main

- On appelle `calculerCompteEstBon` avec en paramètre un appel à `genererNombres` (qui retourne la liste des 6 valeurs de départ).
- On affiche le résultat de `calculerCompteEstBon` avec `afficherResultat`.

genererNombres

Cette fonction retourne une liste contenant les 6 valeurs de départ, et récupère le résultat attendu depuis les entrées de l'utilisateur.

Elle renvoie des erreurs si :

- La liste des paramètres entrée par l'utilisateur ne fait pas 7 de long.
- Les 6 valeurs ne sont pas comprises dans l'ensemble de nombres autorisés (1,2,3,4,5,6,7,8,9,10,25,50,75,100).
- Le résultat attendu n'est pas compris entre 101 et 999.

calculerCompteEstBon

- On trie la liste en paramètre, puis on crée un tableau des couples avec `calculerCouples()`.
- pour chaque opération de chaque couple (et tant que on n'a pas atteint `compteEstBon`) :
 - on réalise l'opération courante entre les membres du couple courant (méthode `calculer()`).
 - Si `calculer` renvoie un nombre positif :
 - On met à jour le tableau courant des calculs effectués en position 6-(nombre de valeurs) avec le calcul courant
 - Si l'opération courante est plus proche du résultat attendu que `nbProche` (ancienne opération la plus proche du résultat), on met à jour `nbProche` et on enregistre le tableau courant des calculs effectués dans le tableau des calculs effectués.
 - Si l'opération courante est égale au résultat attendu, on ajoute l'opération dans la liste des opérations menant au compte bon, et on **retourne vrai**.
 - Sinon, si le tableau des valeurs comprend plus de deux valeurs, on appelle récursivement `calculerCompteEstBon()` avec en paramètre la liste des nombres à laquelle on aura enlevé les membres du couple courant, et ajouté le résultat de l'opération courante. Le booléen `compteEstBon` prends la valeur de retour de l'appel récursif, et si `compteEstBon` est vrai on ajoute l'opération dans la liste des opérations menant au compte bon.
- A la fin de la fonction, on **retourne compteEstBon**.

calculerCouples

Cette méthode génère les couples possibles pour la liste `valeurs` passée en paramètre. Cette méthode est optimisée pour ne retourner que (2 parmi `valeurs`) couples au maximum. Elle élimine aussi les couples contenant des valeurs nulles, car ces valeurs ne font pas avancer l'algorithme.

calculer

`calculer` prends en paramètre deux valeurs (a et b) et une opération. Elle retourne l'opération ou -1 si l'opération ne fait pas avancer l'algorithme. C'est cette fonction que nous avons optimisé autant que possible afin de diminuer le nombre d'itérations de `calculerCompteEstBon()`.

afficherResultat

Affiche le résultat du Compte est bon, et la liste des calculs menant au bon compte ou à la valeur la plus proche.