Change-making problem

Elżbieta Kukla, Teresa Ponikowska, Michał Kijowski

29 marca 2012

Articles

- Jeffrey Shallit (Department of Computer Science, University of Waterloo, Ontario, Canada, May 26, 2003)
  What This Country Needs is an 18 cent Piece
- Anna Niewiarowska, Michał Adamaszek,
  Combinatorics of the change-making problem

Introduction

Coinage system

The sequence

$$D = (e_1, e_2, \ldots, e_k), \text{ where } 1 = e_1 < e_2 < \ldots < e_k$$

will be called a **currency** or **coinage system**.

For any amount n we denote by:

- $opt(n; e_1, e_2, \ldots, e_k)$ – the minimal number of coins needed to pay n
- $grd(n; e_1, e_2, \ldots, e_k)$ – the number of coins used when paying n greedily

Average number of coins needed to give change

Cost

Solving the optimal denomination problem for D denominations up to the limit L means determining the denominations $e_1, e_2, \ldots, e_D$ which minimize

$$cost(L; e_1, e_2, \ldots, e_D) := \frac{1}{L} \sum_{0 \leqslant i < L} opt(i; e_1, e_2, \ldots, e_D)$$

Optimal denominations

Optimal denominations of size D and their costs

D	$(e_1, \ldots, e_D)$	$cost(100; e_1, \ldots, e_D)$
1	(1)	49.5
2	$(1, 10), (1, 11)$	9
3	$(1, 12, 19)$	5.15
4	$(1, 5, 18, 25), (1, 5, 18, 29)$	3.89
5	$(1, 5, 16, 23, 33)$	3.29
6	$(1, 4, 6, 21, 0, 37), (1, 5, 8, 20, 31, 33)$	2.92
7	$(1, 4, 9, 11, 26, 38, 44)$	2.65

USA

Denominations in USA

$(e_1, \ldots, e_D)$	$cost(100; e_1, \ldots, e_D)$
$(1, 5, 10, 25)$	4.7
$(1, 5, 18, 25), (1, 5, 18, 29)$	3.89
$(1, 5, 10, 25, 32)$	3.46
$(1, 5, 10, 25, 50)$	4.2
$(1, 5, 10, 18, 25, 50)$	3.18

Greedy methods

We have a given number N to be represented as a nonnegative integer linear combination of denominations $e_1 < e_2 < \ldots < e_D$.

Greedy algorithm

1. Take as many copies a_D of the largest denomination e_D as possible, so that $a_D e_D \leqslant N$.
2. Then set $N := N - a_D e_D$ and continue the procedure with the remaining smaller denominations.

Computational complexity of the problems

Problems

1. Suppose we are given an amount of change to make, say N, and a system of denominations, $1 = e_1 < e_2 < \ldots < e_D$. How easy is it to compute $opt(N; e_1, e_2, \ldots, e_D)$ or find an optimal representation $N = \sum_{1 \leqslant i \leqslant D} a_i e_i$ i.e. one which minimizes $\sum_{1 \leqslant i \leqslant D} a_i$.
   NP-hard problem
2. Suppose we are given N and a system of denominations. How easy is it to determine if the greedy representation for N is actually optimal?
   co-NP-complete problem

Computational complexity of the problems

Problems

3. Suppose we are given a system of denomination. How easy is it to decide whether the greedy algorithm always produces an optimal representation, for all values of N?
   This problem can be solved efficiently! (Pearson Test)
4. Frobenius problem
   We are given a set of D denominations $e_1 < e_2 < \ldots < e_D$ with $gcd(e_1, e_2, \ldots, e_D) = 1$ and we want to find the largest integer N which cannot be expressed in the form $\sum_{1 \leqslant i \leqslant D} a_i e_i$ with the a_i non-negative integers.
   NP-hard problem