COIN CHANGING

Algorithm Poster

Usman Institute Of Technology

Problem

Finding the change of total amount of money using greedy algorithm.

Detail

1. Begin.
2. Take a input for a amount of money from a user.
3. In a array, take a input for different coin which are available for particular amount.
4. Now we select coins according to greedy algorithm.
5. Extract out the coin that is nearest to that particular amount.
6. Subtract the coin from particular amount.
7. Like 140-100 = 40.
8. Next step, will be to find nearest number to 40.
9. Then step 6 and 7 will repeat until all the amount convert into change.
10. Done.

Algorithm

Greedy ALGORITHM

(x, c1, c2, …, cn)

SORT n coin denominations so that c1 < c2 < .. < cn

S ← φ

WHILE x > 0

k ← largest coin denomination ck such that ck ≤ x

IF no such k, RETURN "no solution"

ELSE

x ← x – ck

S ← S ∪ { k }

RETURN S . [1]

Example

Example 1:

Is greedy algorithm for any set of denominations?

Answer:

No. Coins/notes available:

1, 10, 21, 34, 70, 100, 350, 1225, 1500.

Greedy algorithm:

140$ = 100 + 34 + 1 + 1 + 1 + 1 + 1 + 1.

Optimal:

140$ = 70 + 70.[1]

Example(Continued)

Example 2:

Given a value V, if we want to make change for V Rs, and we have infinite supply of each of the denominations in Indian currency, i.e., we have infinite supply of { 1, 2, 5, 10, 20, 50, 100, 500, 1000} valued coins/notes, what is the minimum number of coins and/or notes needed to make the change?

Input: V = 70

Output: 2

We need a 50 Rs note and a 20 Rs note.[1]

Result

Getting the total amount of money and having the exact solution as we expected.

References

1. https://www.cs.princeton.edu/courses/archive/spring13/cos423/lectures/04GreedyAlgorithmsI-2x2.pdf