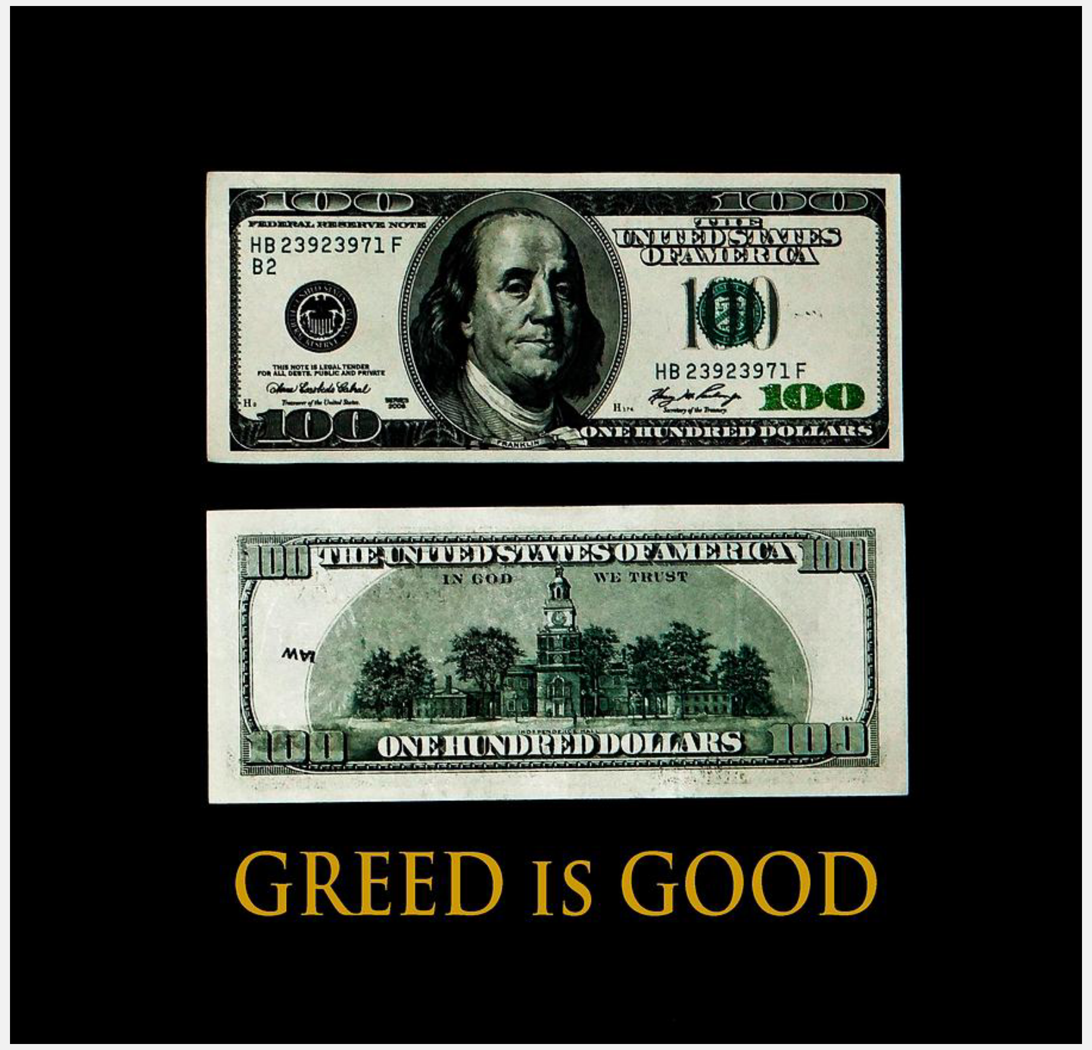
Greedy algorithms

Make locally optimal, irrevocable, choices at each step.

Familiar examples.

- Prim's algorithm. [for MST]
- Kruskal's algorithm. [for MST]
- Dijkstra's algorithm. [for shortest paths]
- Huffman's algorithm. [for data compression]



More classic examples.

- A* search algorithm.
- Gale–Shapley algorithm for stable marriage.
- Greedy algorithm for matroids.
- ...

Caveat. Greedy algorithms rarely lead to provably optimal solutions.

[but often used anyway in practice, especially for intractable problems]

Coin changing problem and cashier's algorithm

Goal. Given U. S. coin denominations $\{ 1, 5, 10, 25, 100 \}$,
devise a method to pay amount to customer using fewest coins.

Ex. 34¢.

6 coins

Cashier's (greedy) algorithm. Repeatedly add the coin of the largest value that does not exceed the remaining amount to be paid.

Ex. $2.89.

10 coins

Algorithm design: quiz 2

Is the cashier's algorithm optimal for U.S. coin denominations { 1, 5, 10, 25, 100 } ?

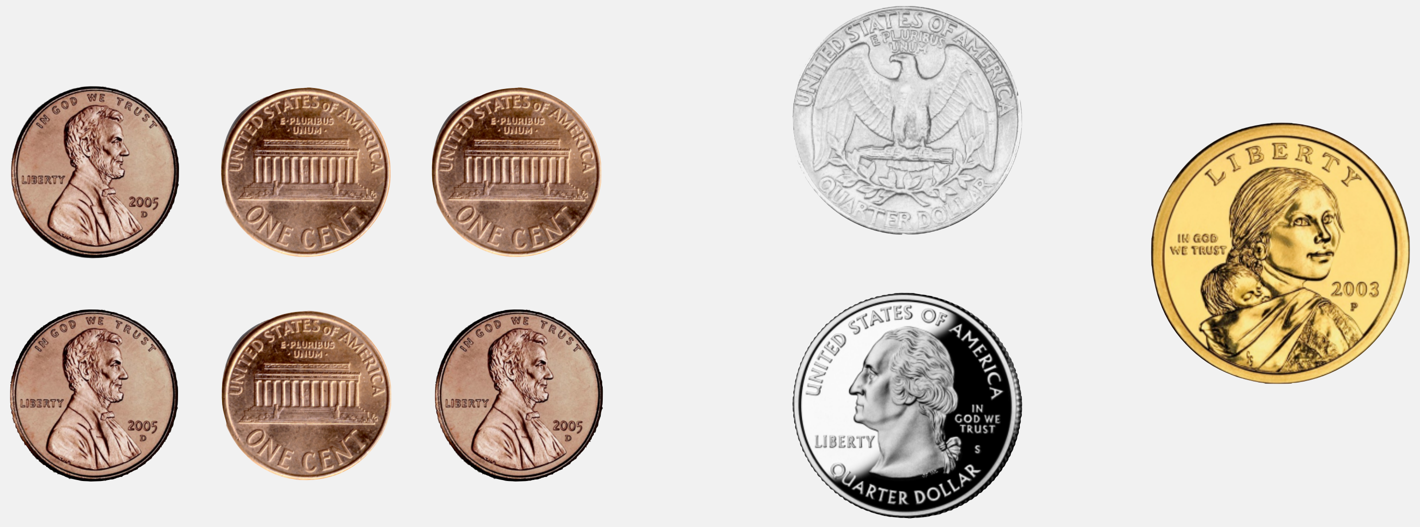
A. Yes, greedy algorithms are always optimal.

B. Yes, for any set of coin denominations $d_1 < d_2 < \ldots < d_n$ provided $d_1 = 1$.

C. Yes, because of special properties of U.S. coin denominations.

D. No.

Properties of any optimal solution (for U.S. coin denominations)



Property 1. Number of pennies $P \leq 4$.

Pf. Replace 5 pennies with 1 nickel.

← exchange argument

Property 2. Number of nickels $N \leq 1$. ← replace 2 nickels with 1 dime

Property 3. Number of dimes $D \leq 2$. ← replace 3 dimes with 1 quarter and 1 nickel

Property 4. Number of quarters $Q \leq 3$. ← replace 4 quarters with 1 dollar

Property 5. $N + D \;\leq\; 2$.

Pf.

- Properties 2 and 3 $\Rightarrow$ $N \leq 1$ and $D \leq 2$.
- If $N = 1$ and $D = 2$, replace with 1 quarter.

Property 6. $\overbrace{P \;+\; 5N \;+\; 10D \;+\; 25Q}^{\text{significance: total amount of change from pennies, nickels, dimes, and quarters}} \;\leq\; 99.$

P1 ⇒ contributes at most 4

P5 ⇒ contributes at most 20

P4 ⇒ contributes at most 75

Optimality of cashier's algorithm (for U.S. coin denominations)

Proposition. Cashier's algorithm yields unique optimal solution for denominations $\{ 1, 5, 10, 25, 100 \}$.

Pf. [for dollar coins]

- Suppose we are changing amount $\$x.yz$.
- Cashier's algorithm takes x dollar coins.
- Suppose (for the sake of contradiction) that an optimal solution takes fewer than x dollar coins.
- Then, optimal solution satisfies $P + 5\,N + 10\,D + 25\,Q \;\geq\; 100$.
- This contradicts Property 6:

$$\boxed{P \;+\; 5\,N \;+\; 10\,D \;+\; 25\,Q \;\;\leq\;\; 99}$$

must make change for ≥ 100¢
using only pennies, nickels, dimes, and quarters

[similar arguments justify greedy strategy for quarters, dimes, and nickels]