Interview Cake

← course home (/table-of-contents)

Greedy Algorithms

A **greedy** algorithm builds up a solution by choosing the option that looks the best at every step.

Say you're a cashier and need to give someone 67 cents (US) using as few coins as possible. How would you do it?

Whenever picking which coin to use, you'd take the highest-value coin you could. A quarter, another quarter, then a dime, a nickel, and finally two pennies. That's a *greedy* algorithm, because you're always *greedily* choosing the coin that covers the biggest portion of the remaining amount.

Some other places where a greedy algorithm gets you the best solution:

- Trying to fit as many overlapping meetings as possible in a conference room? At each step, schedule the meeting that *ends* earliest.
- Looking for a minimum spanning tree in a graph (/concept/graph)? At each step, greedily pick the cheapest edge that reaches a new vertex.

Careful: sometimes a greedy algorithm *doesn't* give you an optimal solution:

- When filling a duffel bag with cakes of different weights and values (/question/cake-thief), choosing the cake with the highest value per pound doesn't always produce the best haul.
- To find the cheapest route visiting a set of cities, choosing to visit the cheapest city you haven't been to yet doesn't produce the cheapest overall itinerary.

Validating that a greedy strategy always gets the best answer is tricky. Either prove that the answer produced by the greedy algorithm is as good as an optimal answer, or run through a rigorous set of test cases to convince your interviewer (and yourself) that its correct.

⬅ course home (/table-of-contents)

Next up: Apple Stocks ➡ (/question/stock-price?course=fc1&section=greedy)