POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

**Strategy 1:** consider items in decreasing order of profit
(i.e., we maximize the local evaluation criterion $p_i$)

Let’s try an example input
- Profits $P = [20, 50, 100]$
- Weights $W = [10, 20, 10]$
- Weight limit $M = 10$

Algorithm selects last item for 100 profit
- Looks optimal in this example

POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

**Strategy 1:** consider items in decreasing order of profit
(i.e., we maximize the local evaluation criterion $p_i$)

How about a second example input
- Profits $P = [20, 50, 100]$
- Weights $W = [10, 20, 100]$
- Weight limit $M = 10$

Algorithm selects last item for 10 profit
- Not optimal!
POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

Strategy 2: consider items in increasing order of weight
(i.e., we minimize the local evaluation criterion \(w_i\))

Counterexample
- Profits \(P = [20,50,100]\)
- Weights \(W = [10,20,10]\)
- Weight limit \(M = 10\)
Algorithm selects first item for 20 profit
- It could select half of second item, for 25 profit!

POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

Strategy 3: consider items in decreasing order of weight divided by weight
(i.e., we maximize local evaluation criterion \(p_i/w_i\))

Let's try our first example input
- Profits \(P = 20, 50, 100\)
- Weights \(W = [10, 20, 10]\)
- Weight limit \(M = 10\)
Profit divided by weight
- \(P/W = [2, 2.5, 1]\)
Algorithm selects last item for 100 profit (optimal)

POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

Strategy 3: consider items in decreasing order of profit divided by weight
(i.e., we maximize local evaluation criterion \(p_i/w_i\))

Let's try our second example input
- Profits \(P = 20, 50, 100\)
- Weights \(W = [10, 20, 10]\)
- Weight limit \(M = 10\)
Profit divided by weight
- \(P/W = [2, 2.5, 1]\)
Algorithm selects second item for 25 profit (optimal)

POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

INFORMAL FEASIBILITY ARGUMENT
(SHOULD BE GOOD ENOUGH FOR ASSESSMENTS)

Feasibility: all \(x_i\) are in \([0, 1]\) and total weight is \(\leq M\)

Either everything fits in the knapsack, or:
- When we exit the loop, weight is exactly \(M\)
  - Every time we write to \(x_i\), it's either 0, 1 or \((M - \text{weight})/w_i\)
    - Rearranging the latter we get \((M - \text{weight})/w_i < 1\)
    - And weight \(\leq M\), so \((M - \text{weight})/w_i \geq 0\)
    - So, we have \(x_i \in [0, 1]\)
MINOR MODIFICATION TO FACILITATE FORMAL PROOF

```plaintext
MINOR MODIFICATION TO FACILITATE FORMAL PROOF

```
The idea is to show that $Y'$ is feasible, and
profit($Y'$) > profit($Y$).
This contradicts the optimality of $Y'$ and proves that $X = Y'$.
To show $Y'$ is feasible, we show $y'_k \geq 0, y'_j \leq 1$ and weight($Y'$) ≤ $M$.

**FEASIBILITY**

- To show $Y'$ is feasible, we show $y'_k \geq 0, y'_j \leq 1$ and weight($Y'$) ≤ $M$.
- Let's show $y'_k \geq 0$.
  - By definition, $y'_k = y_k - \delta w_k$.
  - So, $y'_k \geq 0$ if $y_k - \delta w_k \geq 0$ iff $\delta \leq y_k w_k$.
  - And $y_k$ and $w_k$ are both positive.
  - Therefore, it is possible to choose positive $\delta$ s.t. $y'_k \geq 0$.

Now let's show $y'_j \leq 1$.
- By definition, $y'_j = y_j + \delta w_j$.
- So, $y'_j \leq 1$ if $y_j + \delta w_j \leq 1$ iff $\delta \leq (1 - y_j) w_j$.
- Recall $y_j < x_j$ so $y_j < 1$, which means $(1 - y_j) > 0$.
- So, this constrains $\delta$ to be smaller than some positive number.

Finally, we show weight($Y'$) ≤ $M$.
- Recall changes to get $Y'$ from $Y$.
  - We move $\delta w$ weight from item $k$ to item $j$.
  - This does not change the total weight.
  - So weight($Y'$) = weight($Y$) ≤ $M$.
  - Therefore, $Y'$ is feasible!

**OPTIMALITY**

- Finally we compute profit($Y'$).
- profit($Y'$) = profit($Y$) + $\delta \left( \frac{p_k - p_j}{w_k - w_j} \right)$.
- Since $j$ is before $k$, and we consider items with more profit per unit weight first, we have $\frac{p_j}{w_j} > \frac{p_k}{w_k}$.
- So, if $\delta > 0$ then $\frac{p_k - p_j}{w_k} > 0$.
- Since we can choose $\delta > 0$, we have profit($Y'$) > profit($Y$).

**PROBLEM: COIN CHANGING**

- We can choose $\delta = \frac{\frac{p_k}{w_k} - \frac{p_j}{w_j}}{w_k}$.
- Since $\frac{p_k}{w_k} > \frac{p_j}{w_j}$, we can choose $\delta > 0$.
- Therefore, $X$ is optimal.
EXAMPLE: CANADIAN COINS

Input: coin denominations = 200, 100, 25, 10, 5, 1 (R.I.P.)

Target sum $T = 155$

Output: minimum number of coins to pay $T$

Solution: $1 \times 100 + 2 \times 25 + 1 \times 5$; 4 coins

Suggestion for an algorithm:

- Sort coin denominations from largest to smallest value
- Greedily use the largest possible coin at all times

OPTIMALITY

Is this algorithm optimal?

Trying to build a correctness argument:

- Fix part of the input:
  - Canadian coin system (including pennies)
- Try to prove optimality for all target sums $T$

Reasoning about one class of inputs at a time can make an algorithm easier to understand

We will prove that the greedy algorithm always finds an optimal solution for coin denominations $D = \{100, 25, 10, 5, 1\}$.

We will use the following properties of any optimal solution:

1. The number of pennies is at most 4 (replace five pennies by a nickel)
2. The number of nickels is at most 1 (replace two nickels by a dime)
3. The number of quarters is at most 3 (replace four quarters by a loonie), and
4. The number of nickels + the number of dimes is at most 2 (replace three dimes by a quarter and a nickel; replace two dimes and a nickel by a quarter; the number of nickels is at most one).

The proof is by induction on $T$. As (trivial) base cases, we can take $T = 1, 2, 3, 4$. 

EXAMPLE: CANADIAN COINS (R.I.P. PENNY)
Inductive step (T>4): assume greedy makes optimal change for target values less than T. Show it makes optimal change for T.

Suppose 5 ≤ T < 10. First, assume there is no nickel in the optimal solution. Then the optimal solution contains only pennies, so T ≤ 4 (property (1)); contradiction. Therefore the optimal solution contains at least one nickel. Clearly the greedy solution contains at least one nickel. By induction, the greedy solution for T - 5 is optimal. Therefore the greedy solution for T is also optimal.

(1) the number of pennies is at most 4 (replace five pennies by a nickel)
(2) the number of nickels is at most 1 (replace two nickels by a dime)
(3) the number of quarters is at most 3 (replace four quarters by a loone), and
(4) the number of nickels + the number of dimes is at most 2 (replace three dimes by a quarter and a nickel; replace two dimes and a nickel by a quarter; the number of nickels is at most one).

Recall: properties of any optimal solution

Suppose 10 ≤ T < 25. First, assume there is no dime in the optimal solution. Then the optimal solution contains only nickels and pennies, so T ≤ 5 + 4 = 9 (property (2)); contradiction. Therefore the optimal solution contains at least one dime. Clearly the greedy solution contains at least one dime. By induction, the greedy solution for T - 10 is optimal. Therefore the greedy solution for T is also optimal.

(1) the number of pennies is at most 4 (replace five pennies by a nickel)
(2) the number of nickels is at most 1 (replace two nickels by a dime)
(3) the number of quarters is at most 3 (replace four quarters by a loone), and
(4) the number of nickels + the number of dimes is at most 2 (replace three dimes by a quarter and a nickel; replace two dimes and a nickel by a quarter; the number of nickels is at most one).

Recall: properties of any optimal solution

Exercise: suppose 25 ≤ T < 100
- Find one coin that must be in optimal & greedy to reduce this case to making change for less than T
- Assume no quarters in optimal solution
  - Then by properties 1,4, the optimal solution uses at most: (4 pennies) and (2 nickels or dimes) and (3 quarters)
  - Max value is therefore 99 cents, so cannot make T change!
  - So optimal contains a loone. (And so does greedy.)
  - By inductive hypothesis, greedy is optimal for T - 100.
  - So, greedy is optimal for T.

Recall: properties of any optimal solution

Notice: suppose 25 ≤ T < 100
- Find one coin that must be in optimal & greedy to reduce this case to making change for less than T
- Assume no quarters in optimal solution
  - Then by properties 1,4, the optimal solution uses at most: (4 pennies) and (2 nickels or dimes) and (3 quarters)
  - Max value is therefore 99 cents, so cannot make T change!
  - So optimal contains a loone. (And so does greedy.)
  - By inductive hypothesis, greedy is optimal for T - 100.
  - So, greedy is optimal for T.

Exercise for outside lecture: 200 ≤ T

Exercise: suppose 100 ≤ T < 200
- Find one coin that must be in optimal & greedy to reduce this case to making change for less than T
- Assume no loonies in optimal solution
  - Then by properties 1, 3, 4, the optimal solution uses at most: (4 pennies) and (2 nickels or dimes) and (3 quarters)
  - Max value is therefore 99 cents, so cannot make T change!
  - So optimal contains a loone. (And so does greedy.)
  - By inductive hypothesis, greedy is optimal for T - 100.
  - So, greedy is optimal for T.

Exercise for outside lecture: 200 ≤ T

WHAT ABOUT OTHER COIN SYSTEMS?

Optimal for old Canadian coin system
- Denominations: 200, 100, 25, 10, 5
- Some values can't be created at all!

How about new Canadian coin system
- Denominations: 200, 100, 25, 10, 5
- Some values can't be created at all!

How about the old British coin system
- Denominations: 30, 24, 12, 6, 3, 1
- Counter-example: T=48. Greedy=30,12,6; Opt=24,24

What makes a coin system optimal / non-optimal?
MORE CHALLENGING HOME EXERCISE:

- Show greedy is optimal for any coin system satisfying:
  - \( d_j | d_{j-1} \) for all \( j, 2 \leq j \leq n \)
  - Hints (tiny font, so no spoilers):

  - Is greedy non-optimal for every coin system that does not satisfy this property?
    - No, it’s optimal for old Canadian coins even though 10 does not divide 25

  So, the above condition is sufficient but not necessary