POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

- **Strategy 1**: consider items in *decreasing* order of *profit* 
  (i.e., we maximize the local evaluation criterion $P_j$)
- Let's try an example input
  - Profits $P = [20,30,100]$
  - Weights $W = [10,20,10]$
  - Weight limit $M = 10$
  - Algorithm selects last item for 100 profit
    - Looks optimal in this example
It turns out strategy #3 is optimal…

POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

- **Strategy 2**: consider items in increasing order of weight (i.e., we minimize the local evaluation criterion $w_j$)
  - Counterexample
    - Profits $P = [20, 50, 100]$  
    - Weights $W = [10, 20, 100]$  
    - Weight limit $M = 10$
    - Algorithm selects first item for 20 profit
    - If could select half of second item, for 25 profit!

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

INFORMAL FEASIBILITY ARGUMENT

- 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 $1 - M = weight + w[i] \geq w[i] > M$
- Rearranging the latter we get $(M - weight)/w[i] < 1$
- And weight $\leq M$, so $(M - weight)/w[i] \geq 0$
- So, we have $x_i \in [0, 1)$
MINOR MODIFICATION TO FACILITATE FORMAL PROOF

```
GreedyRationalKnapsack(p[1,...,n], w[1,...,n], W)
X[0,...,n] = 0
for i = 1,...,n
  if weight = w[i] = w
    X[i] = 1
  else
    weight = weight - w[i]
  break
return X
```

Does NOT change behaviour of the algorithm at all!

\[
∀\ i : x'_i \in \{0, 1\}, \text{weight}' = σ_i = 1 \cdot \sum w_i x'_i \leq M
\]

\[
∀\ i : x'_i \in \{0, 1\}, \text{weight}' = σ_i = 1 \cdot \sum w_i x'_i \leq M
\]

OPTIMALITY

For simplicity, assume that the profit / weight ratios are all distinct, so

\[
\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \ldots \geq \frac{p_n}{w_n}
\]

Suppose the greedy solution is \(X = (x_1, \ldots, x_n)\) and the optimal solution is \(Y = (y_1, \ldots, y_n)\).

We will prove that \(X = Y\), i.e., \(x_j = y_j\) for \(j = 1, \ldots, n\). Therefore there is a unique optimal solution and it is equal to the greedy solution.

Suppose \(X \neq Y\) to obtain a contradiction.

Pick the smallest integer \(j\) such that \(x_j \neq y_j\).

\(x_j\) and \(y_j\) are identical up to \(j - 1\), respectively.
fraction of item in knapsack

Greedy solution X

Optimal solution Y

\[ y' = y \]

\[ \Delta y' = y - y \]

\[ \delta y \]

\[ \delta y \]

\[ \delta y \]

\[ \delta y \]

\[ \delta y \]

Since item j is worth more per unit weight than item k, we remove an infinitesimal \( \delta y \) of weight of item k, and add \( \delta y \) weight of item j.

\[ \frac{y}{y'} = \frac{y - \delta y}{y + \delta y} \]

\[ \frac{y}{y'} = \frac{y - \delta y}{y + \delta y} \]

\[ \frac{y}{y'} = \frac{y - \delta y}{y + \delta y} \]

\[ \frac{y}{y'} = \frac{y - \delta y}{y + \delta y} \]

\[ \frac{y}{y'} = \frac{y - \delta y}{y + \delta y} \]

\[ \frac{y}{y'} = \frac{y - \delta y}{y + \delta y} \]
Modified optimal solution $Y'$

The idea is to show that $Y'$ is feasible, and $\text{profit}(Y') > \text{profit}(Y)$. 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($Y'$) $\leq M$.

**FEASIBILITY**

- To show $Y'$ is feasible, we show $y_k' \geq 0, y_j' \leq 1$ and weight($Y'$) $\leq M$.
- Now let's show $y_j' \leq 1$.
  - By definition, $y_j' = y_j + \frac{\delta w_j}{y_j}$.
  - So, $y_j' \leq 1$ iff $y_j + \frac{\delta w_j}{y_j} \leq 1 \iff \delta \leq (1 - y_j)w_j$.
  - 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.

**FEASIBILITY**

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

**OPTIMALITY**

- Finally, we compute $\text{profit}(Y')$.
  - $\text{profit}(Y') = \text{profit}(Y) + \frac{\delta p_j}{w_j} - \frac{\delta p_k}{w_k}$
  - $= \text{profit}(Y) + \delta \left( \frac{p_j}{w_j} - \frac{p_k}{w_k} \right)$.
  - 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 $\delta \left( \frac{p_j}{w_j} - \frac{p_k}{w_k} \right) > 0$.
  - Since we can choose $\delta > 0$, we have $\text{profit}(Y') > \text{profit}(Y)$.

**PROBLEM: COIN CHANGING**
Problem 4.5: Coin Changing

Instance: A list of coin denominations, \( d_1, d_2, \ldots, d_n \), and a positive integer \( T \), which is called the target sum.

Goal: Find an \( n \)-tuple of non-negative integers, \( x = (x_1, \ldots, x_n) \), such that \( T = \sum x_i d_i \), and such that \( x \) is minimized.

In the Coin Changing problem, \( x_i \) denotes the number of coins of denomination \( d_i \), that are used, for \( i = 1, \ldots, n \).

The total value of all the chosen coins must be exactly equal to \( T \). We want to minimize the number of coins used, which is denoted by \( x \).

Example: Canadian Coins (R.I.P. Penny)

- Input: coin denominations = 200, 100, 25, 10, 5, 1
- Target sum \( T = 155 \)
- Output: minimum number of coins to pay \( T \) (and list of coins)
- 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 make use of 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 \( T = 1, 2, 3, 4 \).
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 dime 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 dime. Clearly the greedy solution contains at least one dime. 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 dimes is at most 1 (replace two dimes by a dime);  
(3) the number of quarters is at most 3 (replace a quarter and a dime); and  
(4) the number of nickels is at most 2 (replace 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 7 ≤ s ≤ 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 dimes is at most 1 (replace two dimes by a dime);  
(3) the number of quarters is at most 3 (replace four quarters by a dime), and  
(4) the number of nickels is at most 2 (replace two dimes and a nickel by a quarter, the number of nickels is at most one).

Recall: properties of any optimal solution

Exercise: 25 ≤ T ≤ 100

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 dimes is at most 1 (replace two dimes by a dime);  
(3) the number of quarters is at most 3 (replace four quarters by a dime), and  
(4) the number of nickels is at most 2 (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 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 loonie. (And so does greedy.)  
• By inductive hypothesis, greedy is optimal for T−99.  
• So, greedy is optimal for 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 loonie. (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  
• 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=49, 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