Problem 4.4

Knapsack
Instance: Profits $P = \{p_1, \ldots, p_n\}$, weights $W = \{w_1, \ldots, w_n\}$, and a capacity $M$. These are all positive integers.
Feasible solution: A subset $X = \{i_1, \ldots, i_k\}$ where $\sum_{i=1}^{k} w_{i} \leq M$.

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
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!

- **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
    - $P/W = [2, 2.5, 1]$
    - Algorithm selects last item for 100 profit (optimal)

POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

**INFORMAL FEASIBILITY ARGUMENT**

(should be good enough to show feasibility on assessments)

- Feasibility: all $x_i$ are in $[0, 1]$ and total weight $w \leq M$
- 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 $(M - weight)/w_i$ where weight + $w[i] > M$
- 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

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

Does NOT change behaviour of the algorithm at all!

FORMAL FEASIBILITY ARG

- Loop Invariant: \( V_j : x_j \in [0,1] \)
- \( \text{and weight} = \sum_{i=1}^{n} w_i x_i \leq M \)
- Base case, initially weight = 0 and \( V_j : x_j = 0 \),
  \( \text{So} \ 0 = \text{weight} = \sum_{i=1}^{n} w_i x_i \leq M \)
- Inductive step,
  \( \text{Suppose invariant holds at start of iteration } i \)
  \( \text{Let weight}' , x_i' \) denote values of weight, \( x_i \) at end of iteration \( i \)
  \( \text{Prove invariant holds at end of iteration } i \)
  \( \text{i.e., } V_j : x_j \in [0,1] \text{ and weight}' = \sum_{i=1}^{n} w_i x_i' \leq M \)

FORMAL FEASIBILITY ARG

- WTP: \( V_j : x_j \in [0,1] \)
  \[ \text{and weight} = \sum_{i=1}^{n} w_i x_i \leq M \]
- Case 1: weight + \( w_i \leq M \)
  \( x_i = 1 \text{ which is } x_i \in [0,1] \) \( \text{by line 11} \)
  \( \text{weight}' = \text{weight} + w_i \) \( \text{by line 12} \)
  and \( \text{this is } M \text{ by the case} \)
  \( \text{weight}' = \sum_{i=1}^{n} w_i x_i + w_i \) \( \text{by invariant} \)
  \( \text{weight}' = \sum_{i=1}^{n} w_i x_i + x_i w_i \) \( \text{since } x_i = 1 \)
  \( \text{And } x_i' = x_i = x_i \) \( \text{for all } k \neq i \text{ and } x_i = 0 \text{ so } \sum_{k=1}^{n} x_i w_k = x_i w_i + \sum_{k=1}^{n} x_k w_k \)
  \( \text{Rearrange to get } \sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{n} x_i w_i - x_i w_i \)
  \( \text{So weight}' = \sum_{i=1}^{n} x_i w_i - x_i w_i + x_i w_i = \sum_{i=1}^{n} x_i w_i \)

FORMAL FEASIBILITY ARG

- WTP: \( V_j : x_j \in [0,1] \)
  \[ \text{and weight} = \sum_{i=1}^{n} w_i x_i \leq M \]
- Case 2: weight + \( w_i > M \)
  \( \text{We have } w_i > M - \text{weight} \) \( \text{and } M = \text{weight} \geq 0 \)
  \( \text{by case} \)
  \( \text{So } w_i \leq \|M - weight\| < 1 \text{ which means } x_j \in [0,1] \)
  \( \text{weight}' = M = \text{weight} + \text{(M - weight)} \) \( \text{by line 8} \)
  \( \text{weight}' = \sum_{i=1}^{n} w_i x_i + (M - \text{weight}) \) \( \text{by invariant} \)
  \( \text{But } x_i' = x_i \text{ for all } k \neq i \text{ and } x_i = 0 \text{ so } \sum_{k=1}^{n} x_i w_k = x_i w_i + \sum_{k=1}^{n} x_k w_k \)
  \( \text{Rearrange to get } \sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{n} x_i w_i - x_i w_i \)
  \( \text{So weight}' = \sum_{i=1}^{n} x_i w_i - x_i w_i + (M - \text{weight}) \)
  \( \text{And } M = \text{weight} \text{ so weight}' = \sum_{i=1}^{n} x_i w_i \)

OPTIMALITY

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

- \( p_1 / w_1 > p_2 / w_2 > \cdots > p_n / w_n \)

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

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 \( x_j 
eq y_j \).

Then \( x_j \) and \( y_j \) are identical up to \( x_j \) and \( y_j \), respectively.

\( X = Y \) if \( j \) first index where the solutions differ.

What's the relationship between \( x_j \) and \( y_j \)?
Since item j is worth more per unit weight, replacing even a tiny amount of item k with item j will improve the solution.

So, we remove an infinitesimal $\delta > 0$ of weight of item k, and add it weight of item j.

Must exist $\delta > 0$ such that $y' > 0$.

Remove some of item k and replace it with some of item j.

How much of item k should we remove?

To move $\delta$ weight from item k to item j...

What fraction of item j are we adding?

What fraction of item k are we removing?

Since item j is worth more per unit weight, replacing even a tiny amount of item k with item j will improve the solution.

So, we remove an infinitesimal $\delta > 0$ of weight of item k, and add it weight of item j.

Must exist $\delta > 0$ such that $y' > 0$.

Remove some of item k and replace it with some of item j.

How much of item k should we remove?

To move $\delta$ weight from item k to item j...

What fraction of item j are we adding?

What fraction of item k are we removing?

Since item j is worth more per unit weight, replacing even a tiny amount of item k with item j will improve the solution.

So, we remove an infinitesimal $\delta > 0$ of weight of item k, and add it weight of item j.

Must exist $\delta > 0$ such that $y' > 0$.

Remove some of item k and replace it with some of item j.

How much of item k should we remove?

To move $\delta$ weight from item k to item j...

What fraction of item j are we adding?

What fraction of item k are we removing?

Since item j is worth more per unit weight, replacing even a tiny amount of item k with item j will improve the solution.

So, we remove an infinitesimal $\delta > 0$ of weight of item k, and add it weight of item j.

Must exist $\delta > 0$ such that $y' > 0$.

Remove some of item k and replace it with some of item j.

How much of item k should we remove?

To move $\delta$ weight from item k to item j...

What fraction of item j are we adding?

What fraction of item k are we removing?

Since item j is worth more per unit weight, replacing even a tiny amount of item k with item j will improve the solution.

So, we remove an infinitesimal $\delta > 0$ of weight of item k, and add it weight of item j.

Must exist $\delta > 0$ such that $y' > 0$.

Remove some of item k and replace it with some of item j.

How much of item k should we remove?

To move $\delta$ weight from item k to item j...

What fraction of item j are we adding?

What fraction of item k are we removing?

Since item j is worth more per unit weight, replacing even a tiny amount of item k with item j will improve the solution.

So, we remove an infinitesimal $\delta > 0$ of weight of item k, and add it weight of item j.

Must exist $\delta > 0$ such that $y' > 0$.

Remove some of item k and replace it with some of item j.

How much of item k should we remove?

To move $\delta$ weight from item k to item j...

What fraction of item j are we adding?

What fraction of item k are we removing?

Since item j is worth more per unit weight, replacing even a tiny amount of item k with item j will improve the solution.

So, we remove an infinitesimal $\delta > 0$ of weight of item k, and add it weight of item j.

Must exist $\delta > 0$ such that $y' > 0$.

Remove some of item k and replace it with some of item j.

How much of item k should we remove?

To move $\delta$ weight from item k to item j...

What fraction of item j are we adding?

What fraction of item k are we removing?

Since item j is worth more per unit weight, replacing even a tiny amount of item k with item j will improve the solution.

So, we remove an infinitesimal $\delta > 0$ of weight of item k, and add it weight of item j.

Must exist $\delta > 0$ such that $y' > 0$.

Remove some of item k and replace it with some of item j.

How much of item k should we remove?

To move $\delta$ weight from item k to item j...

What fraction of item j are we adding?

What fraction of item k are we removing?

Since item j is worth more per unit weight, replacing even a tiny amount of item k with item j will improve the solution.

So, we remove an infinitesimal $\delta > 0$ of weight of item k, and add it weight of item j.

Must exist $\delta > 0$ such that $y' > 0$.

Remove some of item k and replace it with some of item j.

How much of item k should we remove?

To move $\delta$ weight from item k to item j...

What fraction of item j are we adding?

What fraction of item k are we removing?
Modified optimal solution $Y'$. 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 $y'_j \geq 0, y'_j' \leq 1$ and weight($Y'$) \leq M.

FEASIBILITY OF $Y'$
- To show $Y'$ is feasible, we show $y'_j \geq 0, y'_j' \leq 1$ and weight($Y'$) \leq M.
- Now let's show $y'_j' \leq 1$.
  - By definition, $y'_j' = y_j' + \frac{\delta}{w_j}$.
  - So, $y'_j' \leq 1$ iff $y_j' + \frac{\delta}{w_j} \leq 1$ iff $\delta \leq (1 - y_j') \frac{w_j}{\delta}$.
- 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 OF $Y''$
- Finally, we show weight($Y''$) \leq M.
  - Recall changes to get $Y''$ from $Y'$.
  - We move a weight from item $j$ to item $k$.
  - This does not change the total weight.
  - So weight($Y''$) = weight($Y$) \leq M.
  - Therefore, $Y''$ is feasible.

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

Task: An instance of non-negative integers, say \( \{x_1, \ldots, x_n\} \), such that \( T = \sum x_i d_i \), and such that \( N = \sum x_i \), 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 \( N \).

EXAMPLE: CANADIAN COINS (R.I.P. PENNY)

- Input: coin denominations \( = 200, 100, 25, 10, 5, 1 \) (R.I.P.)
- 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 \textbf{one class of inputs} at a time can make an algorithm easier to understand
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 7 ≤ T ≤ 9 (property (3)); 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.

Recall: 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 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).

Exercise: 25 ≤ T < 100

Recall: proof for 10 ≤ T < 25

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 ≤ T ≤ 25 (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 = 25 is optimal. Therefore the greedy solution for T is also optimal.

Recall: 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 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).

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 loone. (And so does greedy.)
  • By inductive hypothesis, greedy is optimal for T = 100.
  • 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 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
• How about new Canadian coin system?
  • Denominations: 200, 100, 25, 10, 5
  • Same 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 \( 2 \leq j \leq n \)
  - 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