Knapsack Problem

You're going on a 5 day canoeing trip to Algonquin Park.
You want to pack your knapsack to maximize value and minimize weight.

Given n items, item i has weight w_i and value v_i. Weight limit of knapsack is W. Put items in knapsack, sum of weight $\leq W$, maximize sum of values.

[Notation: find $S \subseteq \{1, \ldots, n\}$, $\sum\{w_i \mid i \in S\} \leq W$ and maximize $\sum\{v_i \mid i \in S\}$.]

Two versions of the problem:

- 0-1 knapsack. Items are indivisible (tent, flashlight)
- fractional knapsack. Can use fractions of items (oatmeal, cheese)

We'll see a dynamic programming algorithm for 0-1 knapsack, but (in some sense) the algorithms is not efficient and the problem is hard.

Today: a greedy algorithm for the fractional knapsack.
Example:

$W = 6$

i	v_i	w_i	v_i/w_i
1	12	4	3
2	7	3	$2\frac{1}{3}$
3	6	3	2

Note: it makes sense to order items by value per weight.

For the 0-1 case, greedy gives item 1, value 12 (nothing else fits) but taking items 2 and 3 gives value 13.
For fractional case, greedy takes item 1, leaving weight of 2 free, so take 2/3 of item 2. Value: $12 + \frac{2}{3} \cdot 7$.

Greedy Algorithm

x_i - weight of item i that we take
free$W$:= W
for $i = 1$ **to** n **do**
  $x_i := \min\{w_i, \text{free}W\}$
  free$W$:= free$W - x_i$
od

Note that the solution will look like:

item	1	2	$\cdots$	j	$j+1$	$\cdots$	n
x_i	x_1	x_2	$\cdots$	x_j	0	$\cdots$	0
	w_1	w_2	...				

use all of items 1 ... j-1

use fraction of item j: $0 < x_j \leq w_j$

use none of items j+1 .. n

Final weight: $\sum x_i = W$ (if $\sum w_i \geq W$)

Final value: $\sum \left(\frac{v_i}{w_i}\right) x_i$

Running time $O(n \log n)$ to sort by v_i/w_i.

Claim: The greedy algorithm gives ~~the~~ an optimal solution to the fractional knapsack problem.

Proof:

$$
\begin{array}{lccccccccc}
\text{greedy solution} & x_1 & x_2 & \cdots & x_{k-1} & x_k & \cdots & x_\ell & \cdots & x_n \\
 & \shortparallel & \shortparallel & & \shortparallel & \not\shortparallel\ \vee & & \wedge & & \\
\text{optimal solution} & y_1 & y_2 & \cdots & y_{k-1} & y_k & \cdots & y_\ell & \cdots & y_n
\end{array}
$$

Suppose y is an optimal solutions that matches x on maximum # of indices, say M. Assume, to arrive at a contradiction, that greedy solution is not optimal: $M < n$. We will show that there exists an optimal solution that matches x on at least $M + 1$ indices.

Let k be the first index where $x_k \neq y_k$.
Then $x_k > y_k$ since greedy maximizes x_k.

Since $\sum y = \sum x = W$, there is a later index $\ell > k$ with $y_\ell > x_\ell$.
Exchange weight Δ of item ℓ for equal weight of item k in optimal solution.

$y'_k \leftarrow y_k + \Delta$
$y'_\ell \leftarrow y_\ell - \Delta$

amount we can add to k

Choose $\Delta \leftarrow \min\{y_\ell - x_\ell, x_k - y_k\}$. Then $x_k = y'_k$ or $x_\ell = y'_\ell$ and $\Delta > 0$.

amount we can move from ℓ

change in value

$$\Delta\left(\frac{v_k}{w_k}\right) - \Delta\left(\frac{v_\ell}{w_\ell}\right) = \Delta\left(\frac{v_k}{w_k} - \frac{v_\ell}{w_\ell}\right)$$

This is non-negative because $\frac{v_k}{w_k} \geq \frac{v_\ell}{w_\ell}$ (we sorted this way)
But y was an optimal solution, so this can't be better.
Therefore it's a new optimal solution that matches x on (at least) one more index (k or ℓ).