Group Work:
- Create alg. to approximately solve knapsack with real costs, capacity.
- Reduce knapsack with real costs to knapsack with integer costs.

Real Knapsack
- \( C_1, \ldots, C_n \) \( \in \mathbb{R}^+ \)
- \( w_1, \ldots, w_n \) \( \in \mathbb{Z}^+ \)
- \( v_1, \ldots, v_n \) \( \in \mathbb{Z}^+ \)

Dynamic Programming
- \( C_1', \ldots, C_n' \) \( \in \mathbb{Z}^+ \)
- \( w_1, \ldots, w_n \) \( \in \mathbb{Z}^+ \)
- \( v_1, \ldots, v_n \) \( \in \mathbb{Z}^+ \)

Reduction

Result: \( S \) will always fit in knapsack, but maybe not optimal.
- Knapsack too small: solutions could be valid in.
- Items too big: original isn't an reduction.

\[ f: O(n) \]
\[ g: O(1) \]

Knapsack: \( O(nC') = O(nMC) \)

Why dependence on \( C \) is good to know!

Runtime:
- \( f: O(n) \)
- \( g: O(1) \)