Optimization problems

An optimization problem generally has two parts:

- An objective function that is to be maximized or minimized.
    - For example, find the route that has the cheapest airfare between Boston and Istanbul.
- A set of constraints (possibly empty) that must be honored.
    - For example, don’t exceed an upper bound on the travel time, or on the number of legs in the trip.

Examples of classic optimization problems

- Shortest path
- Traveling salesman
- Bin packing
- Sequence alignment
- Knapsack

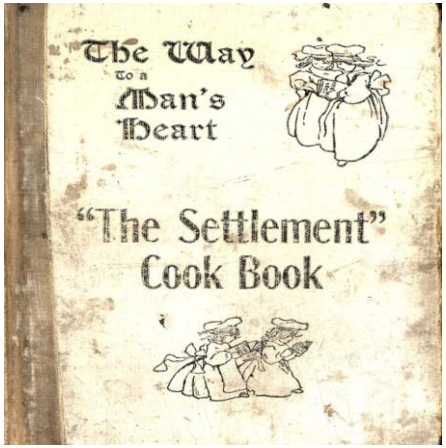
Valuable to know about these kinds of problems, as can use **problem reduction** to reduce a new problem to a variant of a known problem type for which others have created efficient solutions

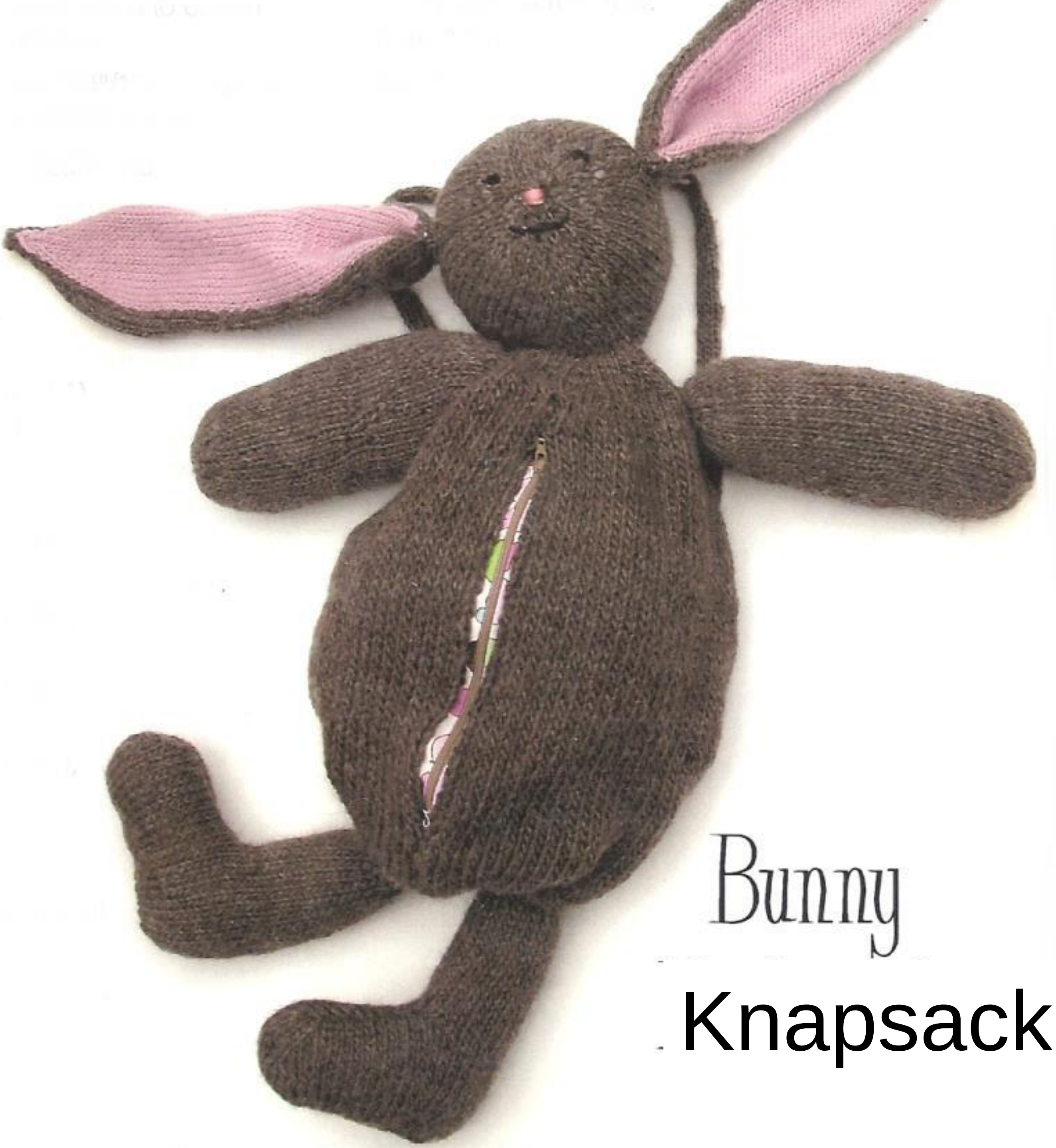
Efficiency of optimization problems

- We’ve seen some very efficient classes of algorithms
  - Sublinear (binary search), linear, polynomial
- Many optimization problems are much worse in efficiency if we want the best answer
- So may want to settle for a “good enough” solution that can be found quickly

A classic example of an optimization problem

- The Knapsack Problem

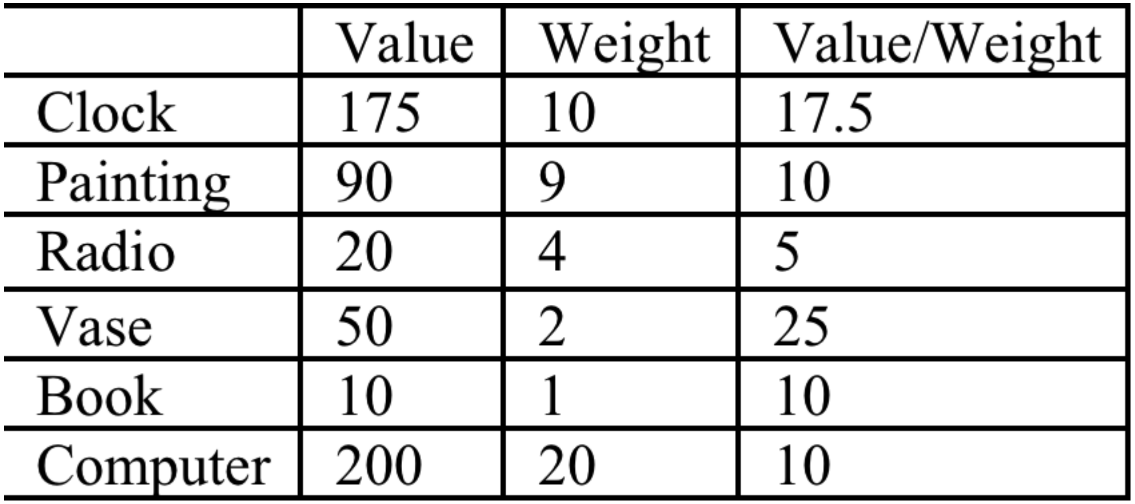




Bunny Knapsack

Some attributes of the “loot”

- Burglar can only carry 20 pounds of loot
- Given this information what should he do?



	Value	Weight	Value/Weight
Clock	175	10	17.5
Painting	90	9	10
Radio	20	4	5
Vase	50	2	25
Book	10	1	10
Computer	200	20	10