A thief enters a house and sees $n$ items; the first item weighs $w_1$ ounces and is worth $v_1$ dollars, the second item weighs $w_2$ ounces and is worth $v_2$ dollars, and so on. Unfortunately (for the thief), he can only carry a total of $W$ ounces in his bag, so he cannot take all the $n$ items in the house. Of course, the thief wants to maximize his profit, thus he must take the set of items that has maximum combined value but whose combined weight does not exceed the capacity of his bag. Since he hasn’t taken Data Structures, the thief needs your help to solve the problem.

Formally, assume that $B = b_1, b_2, ..., b_n$ is a bit vector whose interpretation is $b_i = 1$ if the thief takes item $i$ (which has weight $w_i$ and value $v_i$), and $b_i = 0$ otherwise. Our goal is to find the value of $B$ that maximizes

$$\sum_{i=1}^{n} b_i v_i$$

subject to the constraint

$$\sum_{i=1}^{n} b_i w_i \leq W$$

1. Write a backtracking method that receives $\{w_1, w_2, ..., w_n\}, \{v_1, v_2, ..., v_n\}$, and $W$, and computes $B$. Hint: this problem is NP-complete and is similar to the SubsetSum problem described in class, thus your solution is likely to take $O(2^n)$ time.

2. If each $b_i$ is a floating point number and $0 \leq b_i \leq 1$, meaning that the thief can take fractions of an item if he wants (we can assume that the items are gold bars, for example), the problem becomes easier to solve. Write a greedy method to solve this instance of the problem. Hint: this problem is no longer NP-complete and can be solved in $O(n \log n)$ time.

As usual, write a report describing your work.