Was it worth it?

- This gives a different, and better, answer than any of the greedy methods.
- But how long does it take to find the answer?
- If we have n items, then the power set is the same size as the number of binary numbers with n digits – or 2^n
- Exponential in size if we look at everything
- Even with decision tree approach, still probably exponential

Is it really that bad?

- Suppose we have 50 items
- Suppose we can generate and check each potential combination of these items in a microsecond (one millionth of a second)
- How long will it take to check all combinations?

- About 36 years

The tradeoff

- A greedy algorithm is making the best (as defined by some metric) local choice at each step
- It is making a choice that is **locally optimal**
- But does not mean the solution found is **globally optimal**
- The tradeoff is that an approximation algorithm may find a good, but not best, solution, but take far less time than needed to find the best solution

Can we do better?

- There are few optimizations that might reduce the typical running time. For example, no need to generate overweight solutions.
-
- However, in a theoretical sense, the problem is hopeless. The 0/1 knapsack problem is **inherently exponential** in the number of items.
- In a practical sense, however, the problem is far from hopeless. For example, you might look up the idea of **dynamic programming**.