EC504 Homework Assignment 6 (due December 7 at midnight, online)

1. As a way to relieve some mid-semester stress, you and your friend decide to take a hiking trip in the White Mountains over the weekend. You have packed a number of items for your trip—water, trail mix, emergency kit, extra clothing, etc—and, in the interest of fairness, you want to evenly split the items into the two backpacks that the two of you have brought. Assuming there are n total items, and you know the weight of each item, say, $w_1, ..., w_n$, you want to determine if there is a way to partition the items into the two backpacks so that the sum of the weight of the items in the first backpack *equals* the sum of the weight of the items in the second backpack. Note: this is a special case of the *Subset Sum* problem in which the target number is $W = \frac{1}{2}\sum_{i=1}^{n} w_i$. As we have seen in class, special cases of NP-complete problems are not necessarily themselves NP-complete, but this one is.

   (a) Prove that the *2 Backpack Partition* problem is NP-complete.

   (b) Assuming the weights are positive integers, what conditions on the sum of the w_i would make it possible to solve this problem in polynomial time? Describe such an algorithm. Also give an example input where the running time of this algorithm would not be polynomial time.

2. Your friends' preschool-age daughter Madison has recently learned to spell some simple words. To help encourage this, her parents got her a colorful set of refrigerator magnets featuring the letters of the alphabet (some number of copies of the letter A, some number of copies of the letter B, and so on), and the last time you saw her the two of you spent a while arranging the magnets to spell out words that she knows.

   Somehow with you and Madison, things always end up getting more elaborate than originally planned, and soon the two of you were trying to spell out words so as to use up all the magnets in the full set—that is, picking words that she knows how to spell, so that once they were all spelled out, each magnet was participating in the spelling of exactly one of the words. (Multiple copies of words are okay here; so for example, if the set of refrigerator magnets includes two copies of each of C, A, and T, it would be okay to spell out CAT twice.)

   This turned out to be pretty difficult, and it was only later that you realized a plausible reason for this. Suppose we consider a general version of the problem of *Using Up All the Refrigerator Magnets*, where we replace the English alphabet by an arbitrary collection of symbols, and we model Madison's vocabulary as an arbitrary set of strings over this collection of symbols. The goal is the same as in the previous paragraph.

   Prove that the problem of Using Up All The Refrigerator Magnets is NP-complete.

3. Suppose you are given a set of positive integers $A = \{a_1, a_2, ..., a_n\}$ and a positive integer B. A subset $S \subseteq A$ is called *feasible* if the sum of the numbers in S does not exceed B:

$$\sum_{a_i \in S} a_i \leq B.$$

   You would like to select a feasible subset of A whose total sum is as large as possible.

   Example. If $A = \{8, 2, 4\}$ and $B = 11$, then the optimal solution is the subset $S = \{8, 2\}$.

(a) Here is an algorithm for this problem:

- `Initially` $S = \emptyset$.
- `Define` $T = 0$
- `For` $i = 1, 2, ..., n$
  - `if` $T + a_i \leq B$ `then` $S \leftarrow S \cup \{a_i\}$ `and` $T \leftarrow T + a_i$
- `endfor`

Give an instance in which the total sum of the set S returned by this algorithm is less than half the total sum of some other feasible subset of A.

(b) Give a polynomial-time approximation algorithm for this problem with the following guarantee: it returns a feasible set $S \subseteq A$ whose total sum is at least half as large as the maximum total sum of any feasible set $S' \subseteq A$. Your algorithm should have a running time of at most $O(n \log n)$.

(c) You can use this approximation algorithm to approximately solve the 2 Backpack Partition problem from question 1. Namely, set $B = \frac{1}{2} \sum_{i=1}^{n} w_i$ and determine approximately if half the weight can be put in a single backpack. Code up both the approximate algorithm as well as the exact one from 1b, and compare the two algorithms on a number of randomly-generated datasets of your choosing. Discuss the following issues: running time, the percentage of time the approximate algorithm finds a solution to the 2 Backpack Partition problem when there exists an exact solution, and the average approximation error of the approximation algorithm.