**Python Lab 11a Loading Cargo**

This lab is based on exercise #39 on page 609 in the text:

"Suppose your job is to supervise the loading of two trucks, each of which can carry, at most, fourteen tons. The cargo is a variety of crates whose total weight is twenty-eight tons but whose individual weights vary from crate to crate. The weight of each crate is marked on its side. What heuristic would you use for dividing the crates between two trucks?"

A. Use the following code to start a program that solves this problem:

**#one possible heuristic solution**

**cargo = [9, 7, 3, 4, 5]**

**print (cargo, "has a sum of " , sum(cargo))**

**truck1 = []**

**truck2 = []**

**for crate in cargo:**

**print (crate)**

**if crate + sum(truck1) <= 14:**

**truck1.append(crate)**

**print (truck1, "has a sum of", sum(truck1))**

B. Modify your code as follows: If the crate cannot fit in truck1 check to see if it can fit in truck2. If it can, add it to truck2.

C. By hand, make changes to initial cargo list. Try a list that has more items. Try a list that has fewer items. Always enter a list that has a sum of 28. Are there situations when your heuristic works well? Are there situations when your heuristic works poorly?

D. In Python, you can sort a list by using the command cargo.sort().

Does this help your algorithm be more efficient?

**cargo.sort()**

E. In Python, you can sort a list from high to low by using the command cargo.sort(reverse=True).

Does this help your algorithm be more efficient?

**cargo.sort(reverse=True)**

F. This algorithm found a solution, but it did not find the optimal solution. Talk with a partner about a different solution. Write a different algorithm to fill up the two trucks. Test your algorithm.

G. There are different ways the cargo list can be created. Make up a list that your algorithm gives an optimal solution. Look at one of your classmates' algorithms, and generate a list that their algorithm would not provide an optimal solution.

**Python Lab 11b Loading Cargo with Randomized Lists**

This lab is an extension of Lab 11a.

A. In the previous lab, you entered in the cargo list by hand. In this lab, we will explore how you can generate a list at random. Start by importing the random module and printing a random integer.

**import random**

**print (random.randint(1,28))**

B. Use this idea inside a while loop. Starting with an empty list, keep adding random numbers as long as cargo has a sum of 28 or less.

**cargo = []**

**while (sum(cargo) < 28):**

**crate = random.randint(1, 28)**

**cargo.append(crate)**

**print (cargo, sum(cargo))**

C. This fills up the list, but you will notice that your sum is larger than 28 in most cases.

You can fix this by changing just one line in the above code.

Make the change, so that the cargo list always adds up to 28.

D. Add user input, so that the user can enter in any number for the sum of the cargo. You will need to change the value '14' in your original code, since you will allow each truck to carry half the weight.

E. Sometimes the user may enter an odd number, that does not divide evenly. Write code to handle this situation. There are many ways to do this, so first think about what you want your program to do in this situation.