

ISC-3313 Final Project

Instructor: Marzieh(Tara) Khodaei

Summer 2021

In order to satisfy FSU's computer competency requirement, each student must complete a computer project. This project represents 30% of the final grade for the course. The project involves:

- A short written report describing the project and the program (around 3-5 pages would be fine);
- A working python program, or jupyter notebook, which you have written;
- An oral presentation (about 10 minutes);

Oral presentations are scheduled for **July 27** and **July 29**. You may make your presentation in a number of ways:

- using Powerpoint slides or other programs to display your talk;
- going to the computer if you need to run your program;

Your presentation will be short. You should be sure to include the following important information:

- a simple explanation of the problem you are working on;
- a small example of your problem;
- a description of the algorithm (the ideas) that you are using to solve the problem;
- how you used python to define your problem, determine the solution, and output the results.

Your report and program are due **July 30**. You should submit them to canvas.

I will NOT be grading your report on proper English, or spelling. I DO expect that your report will contain the information, in written form, that I asked you to present in your oral report:

- a simple explanation of the problem you are working on;
- a small example of your problem;
- a description of the algorithm (the ideas) that you are using to solve the problem;
- how you used python to define your problem, determine the solution, and output the results.

For labs and homeworks, I have NOT required you to comment your programs. But, since this program represents your computer competency project, I expect you to try to make the program neat and readable. I expect your program to have some **comments or explanations** that tell me what the important variables are, and how the program is solving the problem. You do not have to explain every single statement in your program, but you should point out the parts that make the computation work.

The following is a list of projects. Please choose one of these projects that you are interested in and submit to canvas on or before **July 1** a 200 word outline for your project. Instead of choosing a project from this list, you may instead think up one on your own, and propose it to me.

Project List

- **Card Counting:** simulate the game of blackjack. (First you need to find the rules. Assume, for instance, that both you and the dealer always keep drawing until you reach 17 or more.) Be sure to use a single deck of cards, and play until you run out of cards. Bet \$10 on every hand, and keep track of your winnings at the end of each deck. Now find out the basic card counting rules: for a given deck, count every face card and 10 that has been played as -1, and all other cards as +1. Whenever the count is negative, bet \$1; if it is positive, bet 10 times the count. Do you see any benefit to a strategy like this?
Reference: http://en.wikipedia.org/wiki/Card_counting
- **The Cheapest Rug:** your new house has beautiful floors, except for one area where there are n random holes. What's the size of the smallest circular rug you can buy that will cover all the holes? This problem is easy for three holes. But if you can solve that problem, you're halfway to a solution of the general problem.
References: de Berg, van Kreveld, Overmars, Schwarzkopf, Computational Geometry, Section 4.7, "Smallest Enclosing Discs", Springer, 2000.
- **The Day Calculator:** how many days old are you? How many days elapsed between the beginning and end of the American Civil War? If you know the calendar dates of two events, you should be able to count the number of days between them, but it's not a trivial task. However, with a little bit of thought, you can work out the formulas you would need to do this. One idea is called the "Julian Day Number", which assigns a unique counting number to every day in history.
References: Lance Latham, Standard C Date/Time Library, RD Books, 1998.
- **Flip Sorting:** sorting a list of numbers is an important task in computing. Suppose you were asked to sort such a list, but the only way you are allowed to modify the list is to reverse the order of the entries in a consecutive part of the list. You can do that as many times as you want to. Can you sort the list? How fast can you sort it? What is the relationship between this operation and genetic variation in animals?
References: Brian Hayes, "Sorting out the genome", American Scientist, Volume 95, Number 5, September-October 2007, pages 386-391.

- **The Gambler:** a man knows that he must pay a debt of \$20,000 on Monday or he is ruined. He only has \$10,000. He goes to Las Vegas for the weekend and proceeds to gamble. If he bets \$10 at a time, and the odds are even, then what are the chances he will reach his goal? On average, how many bets will be made before he wins \$10,000 or loses all his money?
References: Martin Gardner, "Random Walks and Gambling", in The Mathematical Circus.
- **The Partition Problem:** Suppose an art collector has died, and the will requires that the collection be divided up, as equally as possible, between two heirs. We can model this problem by assuming we have a list of n integers, and that we wish to split the list into two parts whose sums are equal, or as close as possible. Can you work out a program that tries to solve this problem? When n is large, a perfect answer might be hard to find; can you think of a simple approach that might come close to solving the problem?
References: Alexander Dewdney, "The Partition Problem" in The Turing Omnibus; Brian Hayes, "The Easiest Hard Problem", American Scientist, Volume 90, Number 2, March-April 2002, pages 113-117.
- **The Solitary Cards:** Suppose we have a deck of 52 cards, and we place them randomly on a table. Can you identify those cards which are not touching any other cards? To make this doable on a computer, assume each card is 5 units wide and 7 units high, that the table is 500 units wide and 500 units high, and that each card stays in its original orientation, not being turned at an angle. Moreover, assume that the lower left corner of the card is always an integer coordinate between (0,0) and (495,493). You should be able to figure out how to place the cards randomly on the table. You should be able to figure out which cards are not touched by any other card, and might be able to create a plot at the end, showing the positions of the cards, with the "solitary" cards highlighted.
- **The Traveling Salesman Problem:** a traveler must plan a trip that visits each city on a list exactly once. The distances between cities are known (but in some cases there might be no direct link from one to another), and the traveler wants to minimize the total distance traveled. For 5 cities, this is easy; for 10 it becomes hard and quickly impossible. Nonetheless, there are ways to guess a good approximation, and ways to try to make an good travel plan even better. Get a list of locations for the 48 US state capitals, and try some of these methods!
References: Charles Van Loan, Daisy Fan, "Insight Through Computing", SIAM, 2010; <http://www.tsp.gatech.edu/>, <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>, David Cook, "In Pursuit of the Traveling Salesman", ebook available through FSU library.