Lab 4 Journal

Day 1

12/14/2020 - 5:53pm - 7:36pm (1h 43min)

I finished problem 1 (DNA), problem 2 (RNA), problem 3 (REVC), and problem 4 (IPRB). I created a stuff.py for things that I might use again so I can just import it. For now I just have a read file function. I found the first three problems simple, but I struggled with the inheritance probability problem. In the end, I drew a diagram to help me understand how the probabilities worked, and after that it was fine. Halfway through I was debating whether or not to brute force it, but I decided not to :)

12/16/2020 - 1:24pm - 2:36pm (1h 12min)

I started work on problem 5 (FIB) on monday, so I wanted to finish it. I was having a lot of trouble, mostly because I read the question wrong, and thought it was talking about individual rabbits instead of pairs. I then did problem 6 (GC), which I found easier than FIB. I renamed stuff.py to resources.py, and added the `read_fasta(file_path)` function to it (which I yoinked from lab 2). I learned how to use the `max()`function to find which sequence had the highest GC content.

12/16/2020 - 5:30pm - 7pm (1h 30min)

Problem 7 (HAMM) was relatively simple, I just used compared each index of the sequences. For problem 8 (PROT) I repurposed the codon dicts from lab 2, painstakingly changing each T to a U for RNA, which I added to the resources module. Problem 9 was similar to what we did in `findORF()` where I used .find to find the index of all the motifs. *note to self, lists in rosalind are strings with spaces. I think doing these problems before lab 2 would really help because it sort of builds up to it. Problem 10 (CONS) looked complicated at first because of the matrices, but it turned out to be not too difficult. One roadblock I had was submitting it to rosalind, and I ended up creating a `write_file()` function in resources and submitting a text file. I think it messed up because of how long each line is.

12/16/2020 - 8:30pm - 10:30pm (2h)

For problem 11 (FIBD), I originally created a rabbit class with an age property. This worked for the sample dataset, but when I downloaded the actual dataset, I ran into a problem where it would take my computer way too much time after I got past 32 months. I ended up using a dictionary to keep track of ages, which is much much faster than millions of objects. Problem 12 (GRPH) was much easier, I just compared every suffix with every prefix. Problem 13 (IEV) seemed super complicated at first, but I realized it was just addition after writing out the probabilities of each genotype. The last one I did (promise I swear this isn't an addiction) was problem 14 (LCSM), which for me was debug heavy. I had an idea of what to do, where it would go through the first sequence and compare every possible substring with all the others, but I left lots of holes to fix which was a little frustrating. Now that I think about it, I should have just put all the problems in one file instead of 14 whoops.

Day 2 - Break!!!

12/21/2020 - 12:00am - 2:22am (2h 22min)

I started the day/night by reorganizing all my modules. I moved everything into functions so if I ever want to call them from somewhere else or copy them all into one file it'll be much easier. I also attempted problem 15 (LIA) for a while (which I started last week), but after staring at it for hours I still have no clue how to do it - I'll probably ask Ms. DeRanek for help after break. Problem 16 (MPRT) made me happy because I got to use APIs. I used some code from a personal project I'm working on, specifically the `status_code_check()` which is just for debugging and a try get block to get responses. I added `get_uniprot()` to resources.py which returns a dictionary of the fasta file.

12/21/2020 - 12:36pm - 5:58pm (5h 22min)

Problem 17 (MRNA) seemed very complicated, but then I realized I just had to mod 100000 at the end. Then I met with my old buddy findORF again. For problem 18 (ORF), I at first tried to redo the whole thing, but I have no clue how I did it last time, and I ended up just copy pasting the code. I had to google how to do problem 19 (PERM), but after understanding the theory i coded it myself. I used a recursive function, which means it calls back itself. Problem 20 (PRTM) was pretty easy, similar to reading codons. I added a dictionary `monoisotopic_mass_dict` to resources. Problem 21 was also pretty okay, the main part is that I used many many for loops to test each letter in the palindrome. For problem 22, I started with `.find()` similar to with ORFs, but I switched to using `.split()` and then adding the list back together.

12/21/2020 - 6:34pm - 7:02pm (28min)

Problem 23 (LEXF) was similar to PERM, and I used a helper function to do it.

Day 3

1/6/2021 - 7:00pm - 9:00pm (2h)

Long time no see! For problem 24 (LGIS) I tried to brute force an answer, but it took too long to find the answer. I tried searching up the answer online, but after reading multiple answers about both recursive and dynamic solutions, I gave up more confused than when I started. I'll probably also have to ask Ms. Deranek about it. My code for problem 25 (LONG) worked for the sample dataset, but not for the actual dataset. I stopped working shortly after this.

Day 4

1/10/2021 - 9:43pm - 11:20pm (1h 37min)

After around 5h worth of troubleshooting, I realized that the problem was that although there could only be one unique superstring, a read could have multiple overlapping reads to it. *note: I found that this was not the case after whoops. Now I have no clue how to do this again.* At this point, for debugging purposes I created a dictionary that had the head as the key and the tail as the value, and there was no key with multiple tails (only one with no tails). I used this dictionary to combine it into a superstring. This solution ended up working.

1/11/2021 - 5:16pm - 7:05pm (1h 51min)

I started working on problem 26 (PMCH), which looked very complicated. I read through the prompt many times without understanding. Eventually, I started by writing a class for the nodes, and then creating a list of all adjacency edges and basepair edges. I didn't know where to go after that and stopped for the night. lie i kept searching things up and reading solutions, and most solutions use a factorial of the number of As and Cs, which I don't really understand. I guess that's another question I'll jam into the 15min office hours I have with her this friday :)