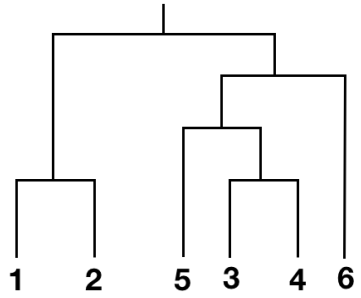
PROBLEM SET 6

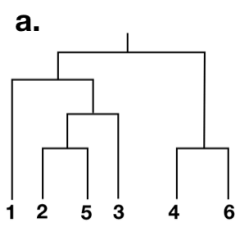
Due Friday, February 21st, by 3:00pm through Canvas. Assignments turned in more than 5 minutes late will be penalized 10 points, with an additional 10 points every 24 hours thereafter.

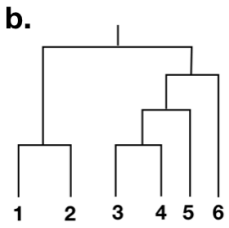
PART A. ALGORITHMS (50 POINTS)

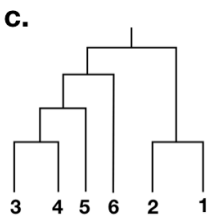
1. (10 points)
Here is a rooted tree representing the evolutionary relationship between six sequences (or organisms):

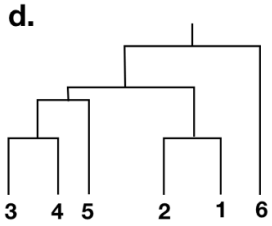


Which of the trees below (if any) are equivalent (represent the same evolutionary relationships) to the tree above? For this problem do not worry about branch lengths (vertical heights of lines), just about the relative evolutionary relationships between sequences. **Choose all that apply**!

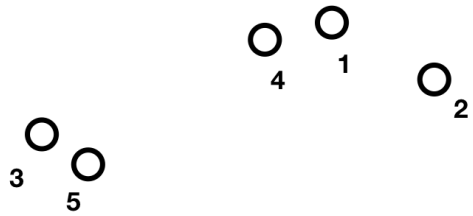








2. (20 points)



a. (10 points) Draw a rooted tree which best represents the distance relationships between points 1-5.

b. (10 points) Which of the distance matrices below best represents the relative distances in your tree? **For each tree you don't choose, please give one example where relative distances between pairs of points are not consistent with your tree.**

A.

	1	**2**	**3**	**4**	**5**
1	0	10	20	6	18
2		0	25	13	22
3			0	17	5
4				0	16
5					0

B.

	1	**2**	**3**	**4**	**5**
1	0	10	20	13	18
2		0	25	6	22
3			0	17	5
4				0	16
5					0

C.

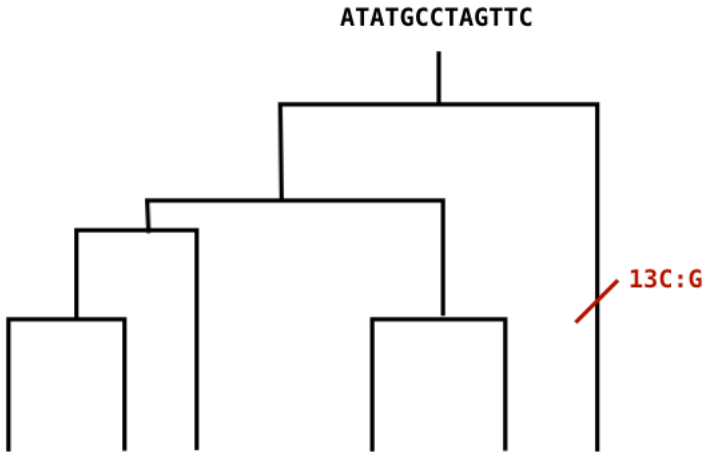
	1	**2**	**3**	**4**	**5**
1	0	20	7	15	2
2		0	4	8	17
3			0	12	13
4				0	16
5					0

D.

	1	**2**	**3**	**4**	**5**
1	0	20	7	8	2
2		0	4	15	17
3			0	12	13
4				0	16
5					0

3. (20 points) Below is a rooted tree representing the six sequences shown below. Fill in the leaves with the numbers 1-6 to make the most parsimonious tree. Then, place every base change at the correct spot on the tree as shown. Your tree should include the base changes: **1A:C, 1A:G, 4T:A, 5G:T, 7C:A, 11T:G, 12T:A, 13C:G**
(13C:G means a C to G transition at base #13)

1. CTAAGCCTAGGTC
2. ATATTCATAGTTC
3. CTATGCCTAGGTC
4. CTATGCCTAGTTC
5. ATATTCCTAGTTC
6. GTATGCCTAGTAG



PART B – PROGRAMMING (50 points)
Please turn in all code as separate .py files.
Allowed packages: sys
Please do not use functions, packages, and syntax (i.e. list comprehensions) we haven't covered in class. The purpose of these problems is to get really comfortable with coding basics. Code which was copied and pasted from the internet will not get credit.

4a. (15 points)
Write a function which takes in two DNA sequences of the same length and outputs the number of differences between them. Turn in a program called **count_different_bases.py** where you (a) define a function, and (b) test that the function does what it is supposed to do. Your program should look something like the structure below. You program does not need to take in user input sequences – you can just define some test sequences within the program.

```
#define function
def <function_name>(<function arguments>):
    …
```

return <number of differences>

#test function
<some code to test that your function is working correctly>

4b. (35 points) Now, let's use the function you just defined to generate a distance matrix between species from a FASTA file (different format than FASTQ from last week). As we learned in class, there are many ways in which we can calculate distances between two sequences. One distance measure is the fraction of base differences over the total number of bases (as in slide #19 from the Phylogenetic Trees lecture from 2/11). Given an input file (**seqs.fasta**) of the format below, write a program which calculates this distance metric between the organisms in the file and prints this matrix to a file called **distance_matrix.txt.** This file should contain tab characters ("\t" in python) between each element and all numbers should be formatted to 2 decimal places (as shown below). Assume all sequences are the same length. Your program should not assume the input file has a specific number of sequences or sequence lengths (get this info within your program from the file itself).

>red_panda
ATCCGTATA
>giant_panda
ATCTGTAAA
>raccoon
ATTTGCAAA
>dog
CTCTGCACA
>wolf
CACTGGACA

You'll run your program with the command:
>python make_distance_matrix.py seqs.fasta distance_matrix.txt

The output file **distance_matrix.txt** should look like this:

```
Organism:       red_panda       giant_panda     raccoon dog     wolf
red_panda       0.00    0.22    0.44    0.44    0.56
giant_panda     0.22    0.00    0.22    0.33    0.44
raccoon 0.44    0.22    0.00    0.33    0.56
dog     0.44    0.33    0.33    0.00    0.22
wolf    0.56    0.44    0.56    0.22    0.00
```

The formatting may look off like it does above, and that's ok. This happens because the species names are different lengths. Not super visually pleasing, but if you were to read this into another program, it would recognize the white space between all values as tabs.

HINTS:
(0) Make a step by step logical plan before you start!
(1) There are many ways to do this problem, but I started by defining a matrix (just a list of lists!) of zeros of the correct size and then filling it in.
(2) If you want to format numbers as you go (turn them into f strings with the correct number of decimals), don't forget that lists in python can contain multiple data types (i.e. ints and strings).
(3) The tab character in python looks like "\t". file.write("string1\tstring2") will output:
string1 string2