15-110 Hw5 - Written Portion

Name:

AndrewID:

#1 - Levels of Concurrency - 8pts

Which of the following concepts does **not** involve running multiple tasks at exactly the same time? Select all answers that apply.

- ☐ Circuit-level concurrency
- ☐ Multitasking
- ☐ Multiprocessing
- ☐ Distributed Computing

For each answer you select, write up to 20 words about what that concept does instead.

#2 - Concurrency Trees - 9pts

Exponentiation (raising a base to a power, like 2^4) can be computed concurrently by first multiplying pairs of bases (e.g., 2*2) together, and then multiplying those products together (4*4), and continuing until there is only one answer (16).

Draw a concurrency tree that computes 2^7 and upload your tree into the next page. You can do this with a picture of a physical drawing or an online image editing tool (like Google Drawings). To upload the image, use the same approach you used on Hw2.

How many **total** steps does this tree take?	
How many **time** steps does this tree take?	

#3 - Parallelization Problems - 4pts

The following scenarios all depict potential problems while parallelizing. Which of these scenarios is the **best** depiction of **deadlock**?

- ☐ Two students are both trying to edit a google doc at the same time, and their text keeps getting mixed up and overlapped.
- ☐ Two students both want to paint the fence at the same time, but one has the paintbrush and the other has the paint bucket, so no painting gets done.
- ☐ Two students are tallying votes in an organization, but they keep erasing and writing over each other's results, so some votes are lost.
- ☐ Two students are trying to split the work of a group assignment, but can't figure out how to split up the problems so that they can be done at the same time.

#4 - Pipelining - 14pts

Note that this problem continues on the next page.

A factory with four workers wants to speed up its production. Currently, each worker makes one shirt at a time by following these steps:

- [S] Set up supplies (for measuring) (5 minutes)
- [M] Measure the fabric (5 minutes)
- [S] Set up supplies (for cutting) (5 minutes)
- [C] Cut out the pattern (5 minutes)
- [S] Set up supplies (for sewing) (5 minutes)
- [W] Sew it all together (5 minutes)
- [F] Fold the shirt (5 minutes).

Note that setup occurs **once** before starting either measuring, cutting, or sewing. When you set up new supplies for a task, you put away the supplies for the previous task.

Each of the cells in the following table represents five minutes, with the whole table representing an hour of work. Fill in the cells with the letters representing the steps to demonstrate the current system the factory uses.

Worker	00:00	00:05	00:10	00:15	00:20	00:25	00:30	00:35	00:40	00:45	00:50	00:55
A												
B												
C												
D												

How many complete, folded shirts could be made by four workers in one hour with the original system?	

Now create a new schedule that uses **pipelining** to increase the efficiency of the shirt-making process. (**Hint:** think about the most efficient way to split up the tasks.)

Worker	00:00	00:05	00:10	00:15	00:20	00:25	00:30	00:35	00:40	00:45	00:50	00:55
A												
B												
C												
D												

How many complete, folded shirts could be made by four workers in one hour using your new pipeline?	

#5 - MapReduce - 10pts

Suppose we implement a MapReduce task to compute how many words in a collection of files have the string "the" in them (e.g., "them" and "whether"). Describe at a high level what **values** this specific mapper function would input and output, what **actions** the collector function would take, and what **values** this reducer would input and output.

Mapper Input Value(s)	
Mapper Output Value(s)	
Collector Actions	
Reducer Input Value(s)	
Reducer Output Value(s)	

#6 - Internet Definitions - 6pts

Fill in the blanks in the following statements with appropriate terms from the Internet lecture.

The internet relies on ___________ to pass packets around the world between computers.

The DNS server translates URLs into ___________, and passes that information back to your computer.

#7 - Internet Communication Process - 4pts

How is a website sent across the internet to your web browser? Select only one answer.

- ☐ The server creates a packet of the whole website and sends it as fast as possible to your browser
- ☐ The server zips up the website to make it as small as possible so none of the data gets lost
- ☐ The server splits the website into packets and makes sure they all get there in the right order
- ☐ The server creates packets and sends them all to your browser, which puts them together in the right order

#8 - Internet Fault Tolerance - 9pts

Which of the following scenarios could lead to you being unable to visit www.cmu.edu? **Select all answers that apply.**

- ☐ Your computer goes down
- ☐ Your ISP goes down
- ☐ A router goes down
- ☐ A DNS server goes down
- ☐ A CMU server goes down
- ☐ All of the CMU servers go down

#9 - Security Definitions - 8pts

Very briefly, and in your own words, what is the difference between authentication and encryption?

#10 - Security Attacks - 8pts

For the two following questions, **select all answers that apply**.

In a Distributed Denial of Service (DDOS) attack, the attacker might do the following to overload the server and make the server stop replying to data requests:

- ☐ Sends a really large message that takes up lots of packets
- ☐ Sends many many different small messages
- ☐ Sends a request for data that doesn't exist on the server
- ☐ Sends a request for a lot of data that will require lots of packets in a response

In a Man in the Middle attack, the attacker:

- ☐ Listens on computer networks for unencrypted usernames and passwords or other data
- ☐ Writes a virus that listens to data passed between your operating system and regular programs
- ☐ Uses social engineering to trick people into sharing their usernames and passwords
- ☐ Puts ads into websites that track everything you do on the website

#11 - RSA Encryption - 6pts

Subra wants to send Farnam a message encrypted using RSA. What step(s) would **Subra** have to take while generating a message to ensure that Farnam can decrypt and read the message?

#12 - Cracking Codes - 6pts

We want to modify the Caesar Cipher algorithm to make it harder to crack. To do this, we double the number of letters that can be substituted in the algorithm by distinguishing between lowercase and uppercase letters. So the entire string of characters we rotate looks like this:

"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"

Does this change the runtime of cracking the Caesar Cipher to be in a new function family? Why or why not?

15-110 Hw5 - Programming Portion

Each of these problems should be solved in the starter file available on the course website. Submit your code to the Gradescope assignment Hw5 - Programming for autograding.

All programming problems may also be checked by running the starter file, which calls the function `testAll()` to run test cases on all programs.

#1 - Counting with MapReduce - 8pts

Implement two functions (a **mapper** and a **reducer**) that can be used in the MapReduce algorithm to count the total number of capitalized words across a collection of files.

For the purposes of this problem, a word is defined as a series of characters separated from other words by either a space or a newline. A capitalized word is one that starts with a capital letter (A-Z), which may be followed by other characters that are or are not capitalized. A word that starts with a non-alphabetic character doesn't count.

First, implement the **mapper**, `mapFileToCount`, which takes a string (text from a file) and returns the number of capitalized words in that string. Note that the text from the file contains many lines of text separated by newlines- you'll need to find a way to split apart words based on both spaces **and** newlines.

We've implemented a **collector** which will call `mapFileToCount` some number of times, then store each of the results in a list.

Second, implement the **reducer**, `reduceCountsToTotal`, which takes a list of integers (the counts combined by the collector) and returns the total number of capitalized words across all the files.

You can test these functions independently by running `testMapFileToCount` and `testReduceCountsToTotal`. When they both work, you can then run `testCountMapReduce` to run the whole MapReduce process using code we provide in `mapreduce.py`.