CMPS 2200 Recitation 02

Name (Team Member 1):__________________________________
Name (Team Member 2):__________________________________

In this recitation, we will investigate recurrences.

Some of your answers will go in `answers.md`. Other prompts will require you to edit `main.py`.

Refer back to the README.md for instruction on git, how to test your code, and how to submit properly to get all the points you've earned.

Recurrences

In class, we've started looking at recurrences and how to we can establish asymptotic bounds on their values as a function of n. In this lab, we'll write some code to generate recursion trees (via a recursive function) for certain kinds of recurrences. By summing up nodes in the recurrence tree (that represent contributions to the recurrence) we can compare their total cost against the corresponding asymptotic bounds. We'll focus on recurrences of the form:

$$W(n) = aW(n/b) + f(n)$$

where $W(1) = 1$.

- ☐ 1. In `main.py`, you have stub code which includes a function `simple_work_calc`. Implement this function to return the value of $W(n)$ for arbitrary values of a and b with $f(n) = n$.

- ☐ 2. (1 points) Test that your function is correct by calling from the command-line `pytest main.py::test_simple_work` by completing the test cases and adding 3 additional ones.

- ☐ 3. (2 points) Now implement `work_calc`, which generalizes the above so that we can now input a, b and a *function* $f(n)$ as arguments. Test this code by completing the test cases in `test_work` and adding 3 more cases.

- ☐ 4. (3 points) Now, derive the asymptotic behavior of $W(n)$ using $f(n) = 1$, $f(n) = n$, and $f(n) = n^2$ with $a = 2$ and $b = 2$. Then, generate actual values for $W(n)$ for your code and confirm that the trends match your derivations.

  Enter your answer in answers.md

- ☐ 5. (4 points) Now that you have a nice way to empirically generate values of $W(n)$, we can look at the relationship between a, b, and $f(n)$. Suppose that $f(n) = n^c$. What is the asymptotic behavior of $W(n)$ if $c < \log_b a$? What about $c > \log_b a$? And if they are equal? Modify `test_compare_work` to compare empirical values for different work functions (at several different values of n) to justify your answer.

  Enter your answer in answers.md

- ☐ 6. (2 points) $W(n)$ is meant to represent the running time of some recursive algorithm. Assume that we always have enough processors for every generated subproblem. Implement the function `span_calc` to compute the empirical span, where the work of the algorithm is given by $W(n)$ and the span of the combine step is equal to the work of the combine step. Implement `test_compare_span` to create a new comparison function for comparing span functions.

- ☐ 7. (3 points) Derive the asymptotic expressions for the span of the recurrences you used in problem 4 above. Confirm that everything matches up as it should.

  Enter your answer in answers.md