LARSON—MATH 353–HOMEWORK WORKSHEET h11
Test 2 Review.

1. Log in to your Sage Cloud account.

   (a) Start the Chrome browser.
   (b) Go to `http://cloud.sagemath.com` and sign in.
   (c) You should see an existing Project for our class. Click on that.
   (d) Click "New", call it **h11**, then click "Sage Worksheet".
   (e) When you are done make a pdf and email that to me.
   (f) This is due *before* the Test 2 start time.

   Definitions.

2. What is a *sufficient* condition? Give an example and explain how it satisfies the definition.

3. What is a *necessary* condition? Give an example and explain how it satisfies the definition.

4. What is a *counterexample* to a mathematical claim? Give an example and explain how it satisfies your definition.

5. What is a *perfect number*? Give examples and explain how they meet the definition.

6. Explain why every perfect number has *abundance index* equal to 2.

7. What is an *abundant* number? Give examples and explain how they meet the definition.

8. In class the CONJECTURING program produced the following conjecture:

   ```
   ((~(has_divisor_deficit))^(has_divisor_surplus))->(is_perfect)
   ```

   (Recall, the caret symbol represents *exclusive or* or XOR). Explain how to *interpret* this conjecture. What does it *mean*? You should explain enough so that it's clear how to check if the conjecture is true or false.

9. I claimed in class that the last conjecture is true. Explain why the conjecture is indeed true. You'll have to dig into the definitions.

10. What is a *graph*? Give an example. List the vertices and edges.

11. What is the *order* of a graph? Give an example.

12. What is the *size* of a graph? Give an example.

13. What is a *triangle-free graph*. Give an example of a triangle-free graph. Explain.

14. What is *Mantel's Theorem*?

15. Explain how Mantel's Theorem can be improved.

Experiments. *The following questions should be answerable by modifying code from our classroom experiments.*

16. **Perfect numbers**. Use the `conjecturing` program to generate *sufficient* conditions for an integer to be perfect.

17. **Perfect numbers**. Use the `conjecturing` program to generate *necessary* conditions for an integer to be perfect.

18. **Navy**. The Navy estimates that there is a 1/3 chance of shooting down an attacking aircraft. What is the probability of shooting it down if 4 missiles are fired?

    Define a function `experiment_navy()` that simulates 4 "shots", and produces and output representing whether or not the missile hit its target. Then repeat this experiment lots of times and calculate an *estimate* of the probability of success.

19. **Coin-tossing**. A trick coin comes up heads 70% of the time. Write some code to simulate 3 coin-flips, repeat this experiment 1000 times, and use this simulation to estimate the probability of getting all 3 heads.

20. **Optimal Stopping**. A coin is flipped n times. Let s_n be the number of Heads that come up. At any point you can stop and collect $\frac{s_n}{n}$ dollars (so the max would be \$1). If you get a Heads on the first flip you should stop. If you get Tails on the first flip, you should flip again.

    Investigate the case where you get Tails on the first flip followed by Heads on the second flip. If you stop now you'll collect $\frac{1}{2}$ dollars.. Should you continue playing? (that is, should you *expect* to get more than $\frac{1}{2}$ dollars?) Write some code to investigate. Explain.

21. **Random Walks**. Define a function `experiment()` that simulates a random walk (your walker should randomly either step one unit to the left or one unit to the right each step) on the number line, starting at the origin, and outputs/returns the number of steps it takes to return to the origin.

22. **Graph Theory**. Use the `conjecturing` program to generate *lower bound* conjectures for the size of a triangle-free graph.

Bonus.

23. **Perfect numbers**. Prove or find a counterexample to one of the sufficient condition conjectures you generated for a number to be perfect.

24. **Perfect numbers**. Prove or find a counterexample to one of the necessary condition conjectures you generated for a number to be perfect.

25. **Coin-tossing**. A trick coin comes up heads 70% of the time. Find the *mathematical* probability of getting all 3 heads.

26. **Graph Theory**. Prove or find a counterexample to one of the *lower bound* conjectures you generated for the size of a triangle-free graph.