CMSC 452 Project 1
DUE March 31 by 11:00am (No dead cat extension)
WARNING: Midterm is 4/2 so you REALLY should do
this much earlier than 3/31 so you have time to study for the midterm.
PLUS you have lots of time to work on the project.
PLUS it is easy—it took Saadiq 10 minutes to do.

1 Small NFAs for L_n

Let L_n be defined as

$$L_n = \{a^i : i \neq n\}.$$

From the notes you know that there are NFAs for L_n of size much smaller than n. How do you find such an NFA? How do you represent it? Here is the procedure:

1. Find (x, y) relatively prime such that $m = xy - x - y \leq n$, but you want to make m not too much smaller than n. We will formalize this as $0 \leq n - (xy - x - y) \leq 4\sqrt{n}$.

2. M is an NFA with (1) a chain of size $c = n - m$ from the start state to a final state q and (2) a loop around q of size $\max(x, y)$. **Note** that q is considered part of the big loop and not part of the chain, but that the start state is considered part of the chain.

3. ε-transitions to states that have loops of size $p_1, \ldots, p_\ell$ (where the p_i's are all prime) where

$$p_1 \times \cdots \times p_\ell \geq n$$

   but you want to keep $p_1 + \cdots + p_\ell$ small. One easy way is to just take the least ℓ such that

$$p_1 \times \cdots \times p_\ell \geq n$$

   and

$$p_1 \times \cdots \times p_{\ell-1} < n.$$

4. On those loops you need to have as final states all states corresponding to $\not\equiv n \pmod{p_i}$.

So the only parameters you need to specify the NFA are

- x, y
- c, the size of the chain
- $p_1, \ldots, p_\ell$ primes

The number of states s in the NFA is $s = \max(x, y) + c + p_1 + \cdots + p_\ell$.

2 Project Requirements

Write a program that will do the following:
Input: $n \in \mathsf{N}$ where $200 \leq n \leq 300$
Output: s, the number of states in the SMALLEST possible small NFA using the procedure outlined above.

1. Your program must accept a SINGLE ARGUMENT (n) in the command line and it must print s and a new line (not return s) to standard output.

2. Your program must be written in Python 3 and called `small_nfa.py`.

3. You may use `numpy`, `argparse`, and `sys` but no other Python libraries.

4. Submit your single `small_nfa.py` file to Gradescope. You have unlimited submissions until the dead cat deadline but we will only run tests on your most recent submission.

5. As a sanity check, the smallest NFA for L_{200} using this procedure has 39 states. So running `python small_nfa.py 200` should yield 39.

6. You may find the `sum`, `prod`, and `sqrt` functions in `numpy` useful.

7. Here is a link to `argparse` documentation: https://docs.python.org/3/library/argparse.html

8. You must submit your own code.