1 Small NFA’s for $L_n$

Let

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

From the notes you know that there are NFA’s for $L_n$ of size much smaller than $n$. How do you find one? How to you represent it?

1. Find $x, y$ such that $m = xy - x - y \leq n$, but you want to make $m$ not to much smaller than $n$.

2. $M_1$ is an NFA with (1) a chain of size $n - m$ from the start state to a state $q$, (2) a loop around $q$ of size $\max(x, y)$,

3. $e$-transitions to states that have loops of size $p_1, \ldots, p_\ell$ 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 $\ell$ such that

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

4. On those loops you need to have as ACCEPTS 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.
- $p_1, \ldots, p_\ell$ and For each $1 \leq i \leq \ell$, you need $m_i \equiv n \pmod{p_i}$.

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

2 Your Project

Write a program that will do the following:

Input: $n \in \mathbb{N}$ and $n \geq 200$.

Output: (what is in parenthesis is not part of the actual output of your program).

- $x, y$ (Must be that $xy - x - y \leq n$ but not too much less. Formally we will require $0 \leq n - (xy - x - y) \leq 4\sqrt{n}$.)
- $c$ (the size of the chain)
- $p_1, \ldots, p_\ell$ (Must have $\prod_{i=1}^{\ell-1} p_i < n$ and $\prod_{i=1}^{\ell} p_i \geq n$.)
- $\max(x, y) + c + p_1 + \cdots + p_\ell$ (the number of states).

Your program must accept a single argument ($n$). Your program must print a single valid json object to standard output, with the following keys and values:

- “x”: $x$
- “y”: $y$
- “c”: size of the chain
- “p”: list of primes $[p_1, \ldots, p_\ell]$
- “s”: total number of states

The first line of your program must be a unix shebang (e.g. `#!/usr/bin/env python3`). You can use Python, Perl, or Ruby (ask us before you use a different language).

As a formatting test, if solution.py is your solution file and test.py is the following python script:

```python
#!/usr/bin/env python3
import sys, json
print(json.load(sys.stdin).keys())
```

Then the following invocation:
$ ./solution.py 200 | ./test.py

Should produce (note that the keys may not appear in this exact order):

dict_keys(['x', 'y', 'c', 'p', 's'])

We’ll be grading your submissions via script, so do not submit your program if this test fails. When you’re ready, submit your code to:
  https://www.dropbox.com/request/0aooO7sDt4Hv8mnXAh1k