$\text{PRIMES} = \{ p \in \mathbb{Z}_+ : p \text{ is a prime} \}$

```
is_prime(p):
    for i in [2, 3, ..., p-1]
        if i divides p
            return False
    return True
```

is_prime() runs in $O(p)$ time

but length of input is $\lceil \log_2 p \rceil$

so this is really an exponential time algo

Today, we design a poly-time verifier for PRIMES

means it runs in $O(\log^d p)$ time

In other words, we show that $\text{PRIMES} \in NP$

$$2^{\log_2 p} = p$$

Warm-up

COMPOSITES $= \{ p : p$ is a composite number $\}$

A verifier for COMPOSITES :

ceritificate $\langle a, b \rangle$ such that $p = a \cdot b$

verifier computes $a \cdot b$ and check it equals p

this can be done in $O(\log^2 p)$ time

<u>Lemma</u> : A odd number p is prime iff

$\exists \; 1 < t < p$:

(i) $t^{p-1} \equiv 1 \pmod{p}$

(ii) $t^{\frac{p-1}{\ell}} \not\equiv 1 \pmod{p}$ $\forall$ prime factor ℓ of $p-1$

First idea:

$$\text{cert}(p) = \langle t, \ell_1, \ell_2, \ldots, \ell_k \rangle$$

where $\ell_1, \ell_2, \ldots \ell_k$ are the prime factors of $p-1$

Example:

$$\text{cert}(11) = \langle 2, 5, 2 \rangle$$

$$2^{10} = 1024 = 93 \times 11 + 1 \equiv 1 \pmod{11}$$

$$2^{2} = 4 = 0 \times 11 + 4 \equiv 4 \pmod{11}$$

$$2^{5} = 32 = 2 \times 11 + 10 \equiv 10 \pmod{11}$$

$$\text{cert}(9) = \langle 8, 8 \rangle$$

$$8^{8} = 1864135 \times 9 + 1 \equiv 1 \pmod{p}$$

$$8^{1} = 0 \times 9 + 8 \equiv 8 \pmod{p}$$

$\uparrow$

The verifier accepts a No instance for a Yes instance

How to solve this problem?

$$cert(p) = \langle t, \ell_1, cert(\ell_1), \ell_2, cert(\ell_2), \ldots, \ell_k, cert(\ell_k) \rangle$$

Now the verifier cannot be fooled but how big is ~~thee~~ certificate and how long dees it take to verify?

<u>Claim</u>: $|cert(p)| = O(\log^2 p)$ and it can be verified in $O(\log^4 p)$ time

<u>Thm</u> [1975 Pratt]: PRIMES $\in NP$

Let $L(p)$ = length of the certificate for p

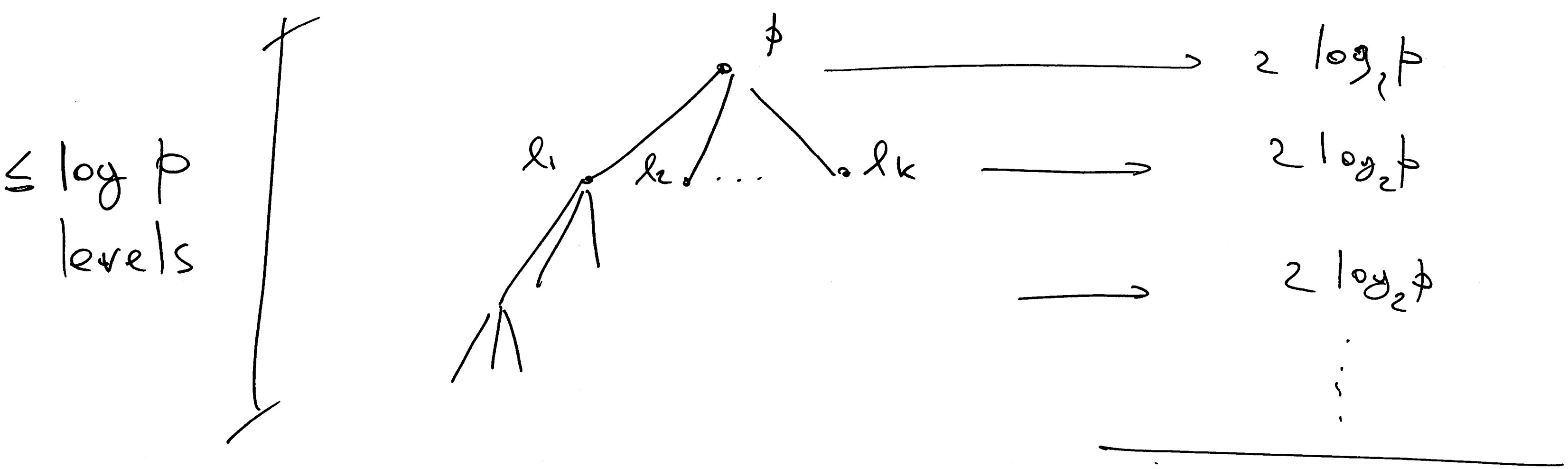
Note that $p \geq l_1 l_2 l_3 \ldots l_k$ so

$$\log_2 p \geq \log_2 l_1 + \log_2 l_2 + \ldots + \log_2 l_k$$

also $p \geq t$ so

$$\log_2 p \geq \log_2 t$$

Thus $L(p) \leq L(l_1) + L(l_2) \ldots + L(l_k) + 2 \log p$



In total $O(\log^2 p)$ bits