

Sample with LuaL^AT_EX

rsk0315

Jun. 17 20:51, 2023 (fa4fb04)

1 Samples

1.1 Mathematical formulæ

$$f(a) = \frac{1}{2\pi i} \cdot \oint_{\gamma} \frac{f(z)}{z-a} dz$$

$$L(\emptyset^* 10^*) = \{\emptyset^m 10^n \mid m, n \in \mathbb{N}\} \in \mathcal{REG}$$

1.2 Pseudocodes

Algorithm 1: 線形篩

```

1 function LINEAR-SIEVE( $n$ )  $\triangleright O(n)$  time
   input:  $n \in \mathbb{N}$ 
   output:  $(\text{lpf}(i))_{i=0}^n \in \mathbb{N}^{n+1}$ 
2    $p \leftarrow ()$   $\triangleright$  a list of primes in ascending order
3    $l \leftarrow (1)_{i=0}^n$   $\triangleright l_i = \text{lpf}(i)$ 
4   for  $i \leftarrow (2, \dots, n)$  do
5     if  $l_i = 1$  then
6        $l_i \leftarrow i$ 
7        $p \leftarrow^{++} (i)$ 
8     for  $j \leftarrow p$  do
9       if  $j > \min\{l_i, \lfloor n/i \rfloor\}$  then break
10       $l_{i \cdot j} \leftarrow j$ 
11  return  $l$ 

```

1.3 Source codes

線形篩

```
struct LinearSieve(Vec<usize>);

impl LinearSieve {
    pub fn new(n: usize) -> Self {
        let mut lpf = vec![1; n + 1];
        let mut primes = vec![];
        for i in 2..=n {
            if lpf[i] == 1 {
                lpf[i] = i;
                primes.push(i);
            }
            let lpf_i = lpf[i];
            for &j in primes.iter().take_while(|&j| j <= lpf_i.min(n / i)) {
                lpf[i * j] = j;
            }
        }
        Self(lpf)
    }

    pub fn dp<T>(
        &self,
        zero: T,
        one: T,
        eq: impl Fn(&T, usize) -> T,
        gt: impl Fn(&T, usize) -> T,
    ) -> Vec<T> {
        let n = self.0.len() - 1;
        if n == 0 {
            return vec![zero];
        } else if n == 1 {
            return vec![zero, one];
        }

        let mut res = vec![zero, one];
        res.reserve(n + 1);
        for i in 2..=n {
            let lpf = self.0[i];
            let j = i / lpf;
            let tmp = if lpf == self.0[j] { eq(&res[j], lpf) } else { gt(&res[j], lpf) };
            res.push(tmp);
        }
        res
    }
}
```

sample.rs

```
use nekolib::math::LinearSieve;

fn main() {
    let ls = LinearSieve::new(10);

    //  $\varphi(\prod_p p^{e_p}) = \prod_p (p-1) \cdot p^{e_p-1}$ 
    let phi = ls.dp(0, 1, |&x, p| x * p, |&x, p| x * (p - 1));
    assert_eq!(phi, [0, 1, 1, 2, 2, 4, 2, 6, 4, 6, 4]);
}
```

1.4 Figures

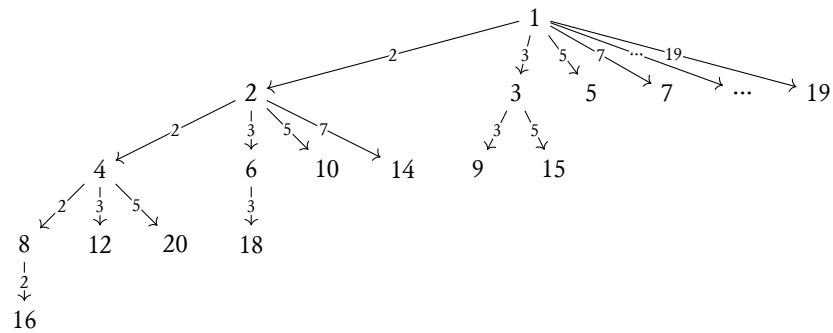


図1 $n/\text{gpf}(n) \rightarrow n$ ($1 < n \leq 20$) に辺がある木。

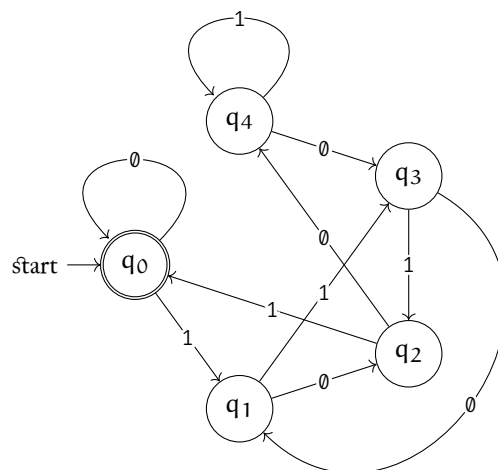


図2 2進法で解釈したときに5の倍数となる0/1列を受理するDFA。

1.5 Hyperlinks

Section 1 [→](#) p. 2

`std::vec::Vec` (<https://doc.rust-lang.org/std/vec/struct.Vec.html>)