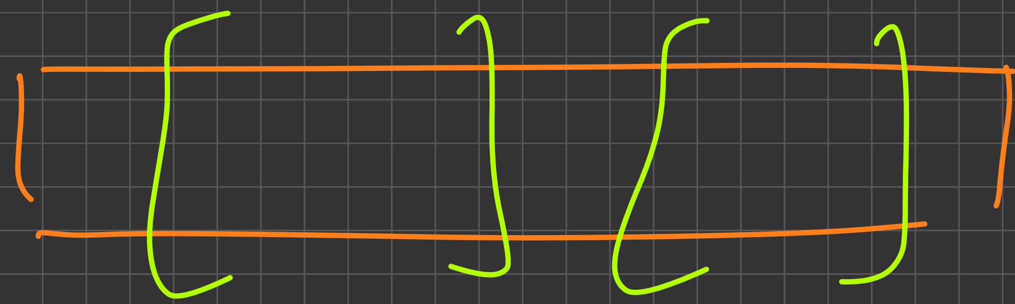
String Hashing

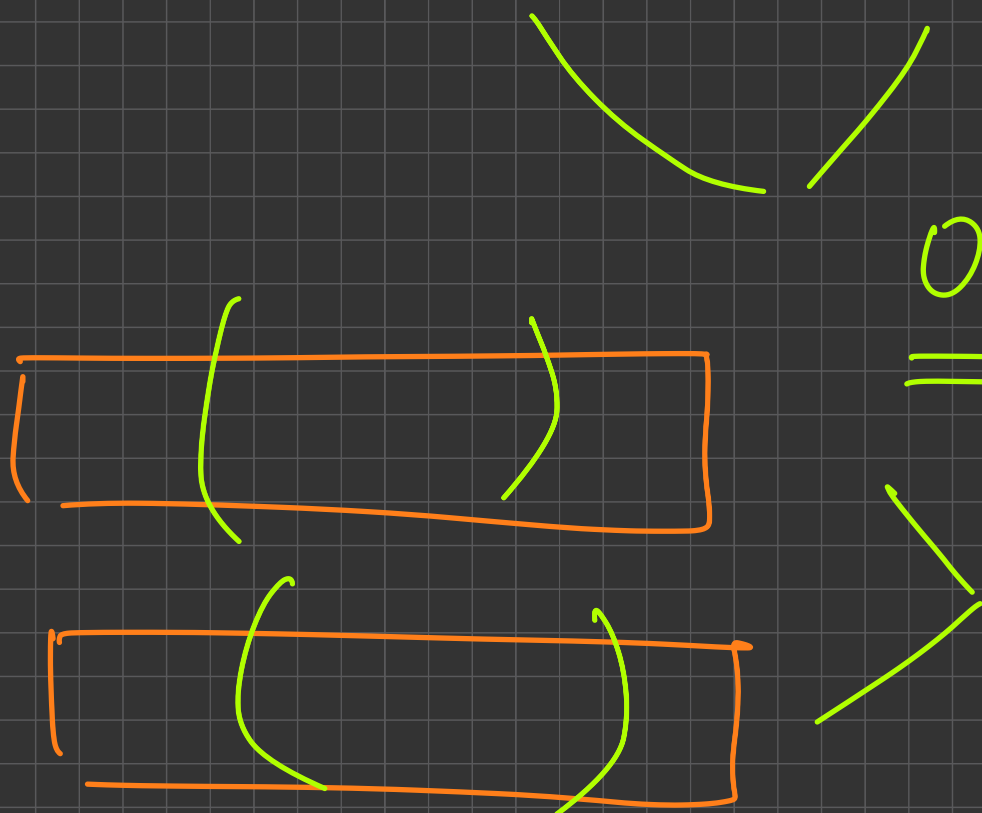
- Priyansh Agarwal

$S =$



$S_1 =$

$S_2 =$

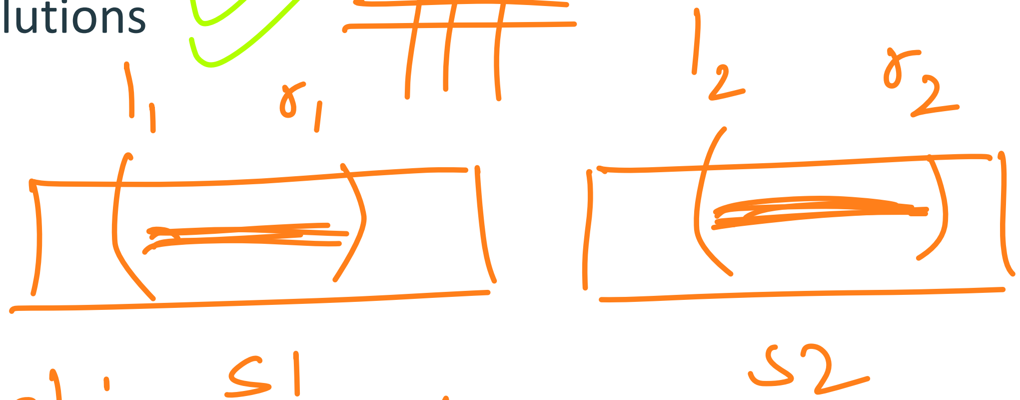


$O(1)$

$O(\log n)$

Why String Hashing?

- Optimizing Brute Force solutions
- Comparing 2 strings
- Daddy of String Algos



string matching

kmp, z, aho-corasick } string hashing

1. "Priyansh" $\longrightarrow$ 1029

2. "TLE" $\longrightarrow$ 729

3. "TLE" $\longrightarrow$ 729

" ______________________ " → 29

" PriyanshTLE123 " → 57

" ______________________ " → 34

" __________________ " → 84

Requirements

- (1) If a == b, Hash(a) == Hash(b)
- If a != b, Hash(a) != Hash(b)
  - If this is true with a very high probability we are good to go
- Calculating Hash function should be fast enough
- (4) Hash of a string shouldn't change in the code
- Hash value should be itself O(1)

"abc" → hash → 29

"bca" → hash → 34

abc → 5

abc → 5

abc → {14, 20, 30, 40}

input → function → output

abc → 29

bca → 34

de → 12

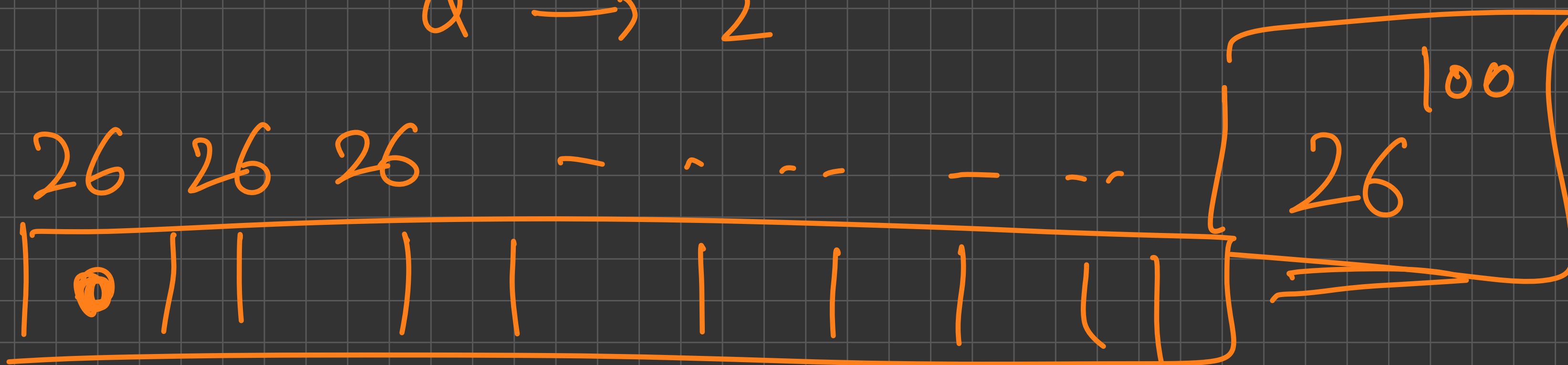
priyansh → 59

abc → 59

priyansh → 59

string s of length 100

'a' $\rightarrow$ 'z'



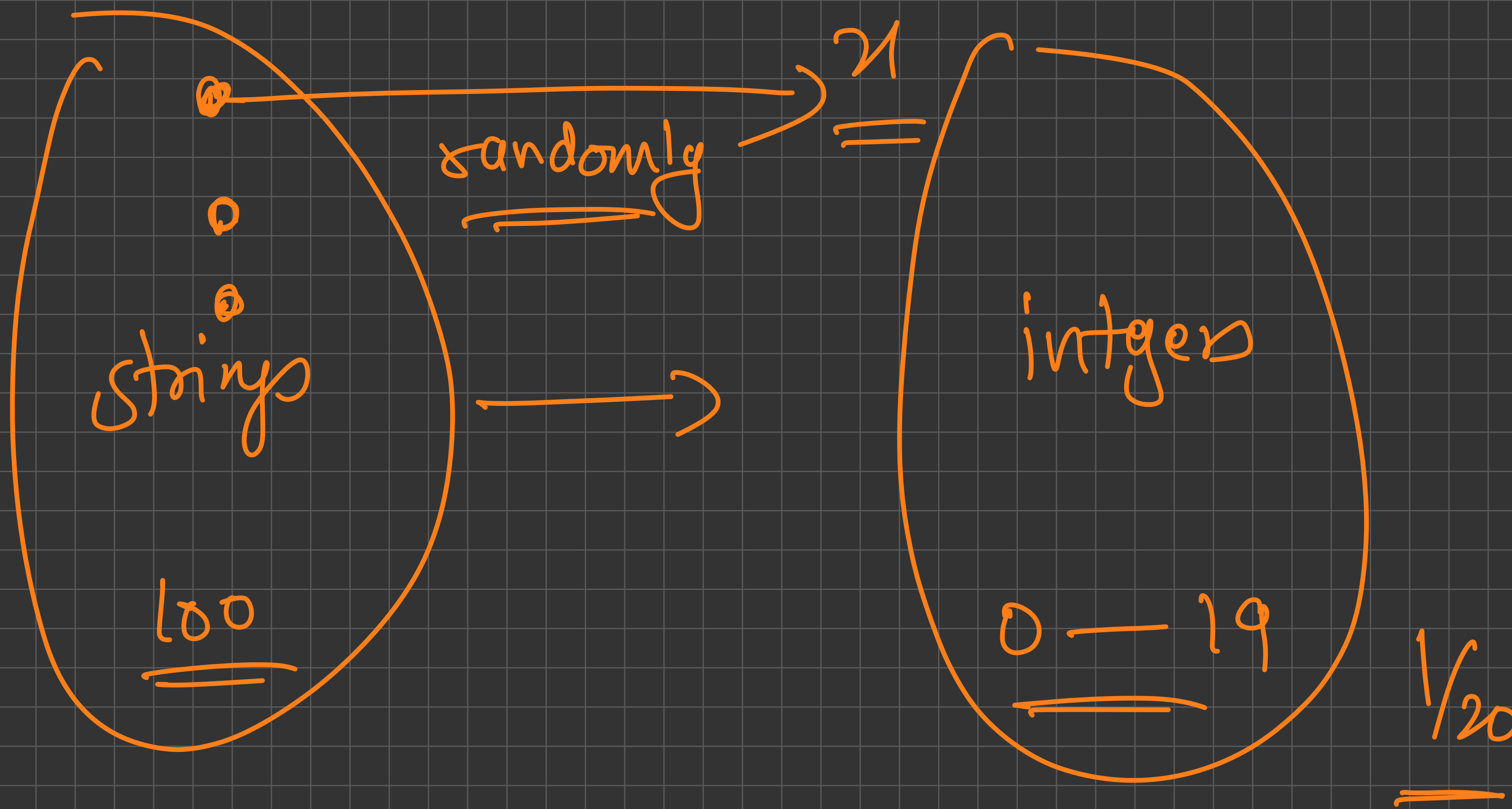
int $\rightarrow$ 10^9

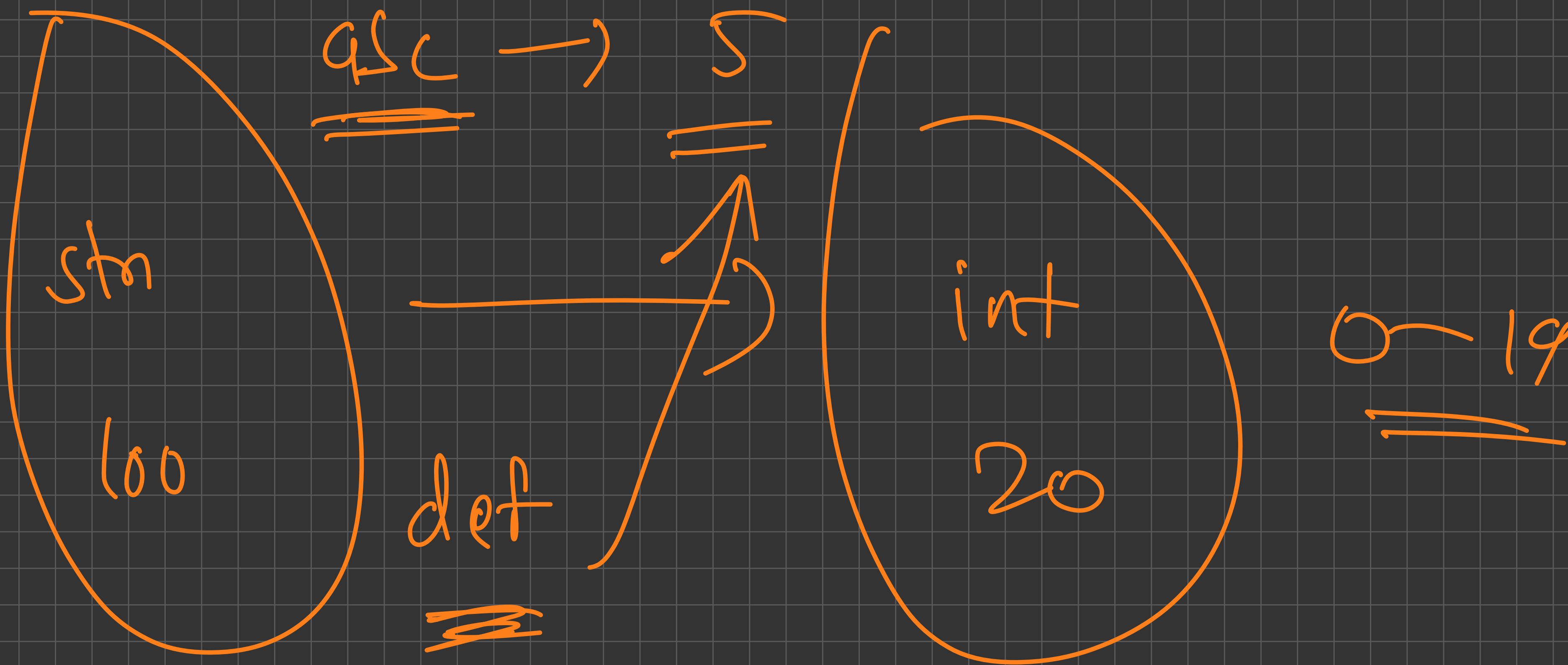
long long $\rightarrow$ $\boxed{10^{18}}$ $\leftarrow$ 26^{100}

10^{100}

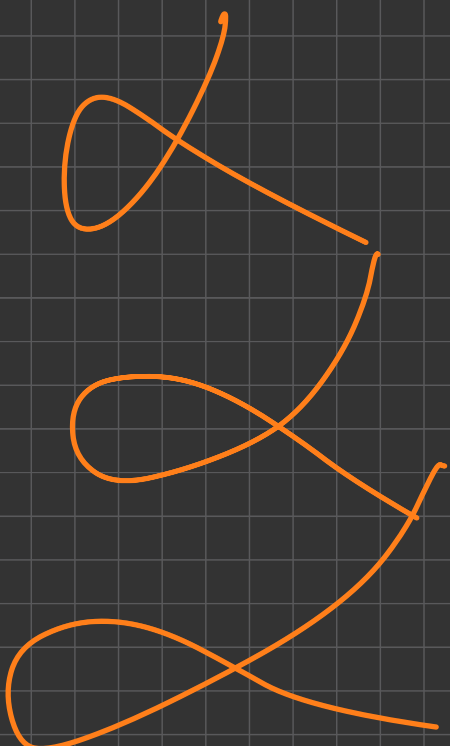
10^5 10^6

26^{100000}

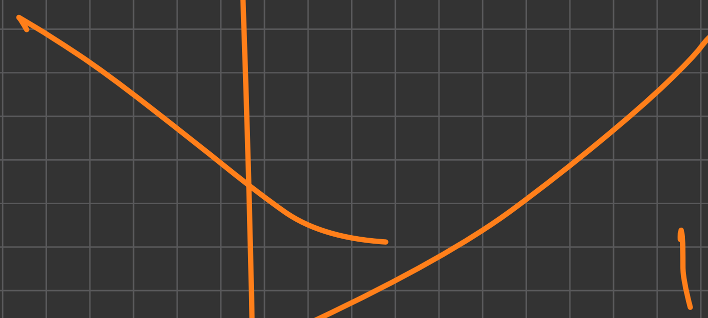

21
randomly
strings
100
integers
0 — 19
1/20


str
100
asc → 5
def
int
20
0-19

abc $\longrightarrow \boxed{5}$



$1/20$



def $\longleftrightarrow$ 5

$1/20$

$1/400$

abc, def $\longrightarrow$ $\underline{\underline{5}}$ $\longrightarrow$ $\underline{\underline{1/400}}$

abc, def $\longrightarrow$ 6 $\longrightarrow$ $1/400$

1

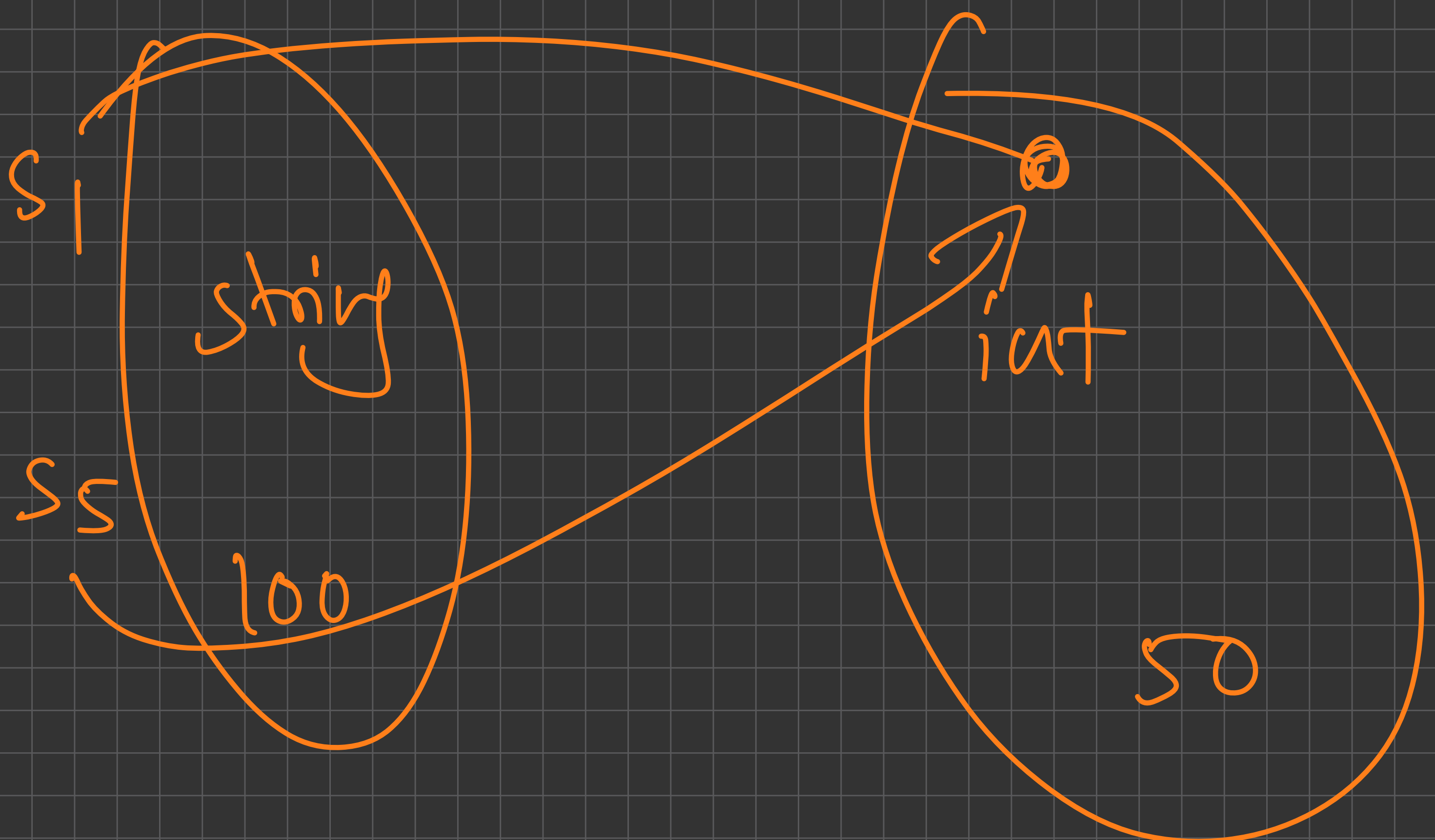
1

1

$\underline{\underline{20}}$

$\frac{1}{400} + \frac{1}{400}$ --- 20 times

$= \underline{\underline{1/20}}$



$\frac{1}{50}$

n strings 2 strings from

m integer 1/m

abc $\longrightarrow$ 0 — 19

abc $\longrightarrow$ 5   1/20

$\longrightarrow$ 6

$\longrightarrow$ 7

$\longrightarrow$ 10

$\longrightarrow$ 19

Polynomial Rolling Hash

def
4 5 6

p = 3
m = 5

$$\text{hash}(s) = s[0] + s[1] \cdot p + s[2] \cdot p^2 + \ldots + s[n-1] \cdot p^{n-1} \mod m$$

$$= \sum_{i=0}^{n-1} s[i] \cdot p^i \mod m,$$

$(4 + 5 \cdot 3 + 6 \cdot 9) \% 5$

agc

$$(s_0 + s_1 \cdot g + s_2 \cdot g^2) \% m$$

$$(s_0 + s_1 + s_2) \% m$$

$$hash(s) = \left[s_0 + s_1 \cdot p + s_2 \cdot p^2 + \cdots + s_{n-1} \cdot p^{n-1} \right] \% \, m$$

'a'

p and m are constant

abc

1 2 3

dea

4 5 1

"aed" $\longrightarrow$ $p = 5$, $\underline{\underline{m = 7}}$

$$hash = (1 + 5 \cdot p + 4 \cdot p^2) \% 7$$

$$= (1 + 25 + 100) \% 7$$

$$= \underset{= 0}{(126)} \% 7 = \underline{\underline{0}}$$

$\underline{\underline{hash}}(s) = \underline{\underline{H}}$

$\Downarrow$

string

$\underline{\underline{p, m}}$

$hash(s + x)$

$\|$

character

$$hash(s) = \left(\underbrace{s_0 + s_1 \cdot p \ldots\ldots s_{n-1} \cdot p^{n-1}}_{H}\right) \% m$$

$$hash(s+x) = \left((\quad) + x \cdot p^{n} \right) \% m$$

$$\hookrightarrow (H + x \cdot p^{n}) \% m$$

$$hash(s) = H$$

$$hash(\underbrace{x}_{\text{character}} + \underbrace{s}_{\text{string}})$$

$$hash(s) = (\underbrace{s_0 + s_1 \cdot p ---- s_{n-1} \cdot p^{n-1}}_{H}) \% m$$

$$\Rightarrow (x + s_0 \cdot p + s_1 \cdot p^2 ---- s_{n-1} \cdot p^{n}) \% m$$

$$= (x + H \cdot p) \% m$$

Why Rolling?

- Hash(s) = H, what is Hash(s + x), x = character
  - $(H + xp^n) \% m$
- Hash(s) = H, what is Hash(x + s), x = character
  - $(x + Hp) \% m$ → $O(1)$

$O(\log n)$

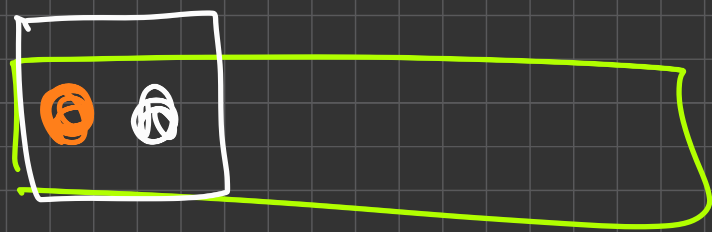
- How about calculating the Hash of a substring quickly?
- How about just comparing two strings?

(asc[dasp]rias) $\Longrightarrow$ $\underline{\underline{O(n)}}$

$\underline{\underline{\qquad\qquad}}$

$\underline{\underline{O(n)}}$

$S \rightarrow$ of length n



s_0

$h[0] = s_0$

$h[1] = (s_0 + s_1 \cdot p)\ \%\ m = (h[0] + s_1 \cdot p)\ \%\ m$

$$(a + b) \% m$$

$$(a - b) \% m$$

$$(a * b) \% m$$

$$(a / b) \% m$$

$$(a/b) \% m$$

$$(a \cdot \text{modulor inverse of } b) \% m$$

$$(a \cdot b^{m-2}) \% m$$

$$a^{p-1} \equiv 1 \mod p$$

$$(a^{p-2}) \% m$$

$$\left(\frac{1}{a}\right) \% \, m =$$

if m is a prime

$$\left(a^{m-2}\right) \% \, m$$

$h[r]$

$l \qquad r$

0 based indexed

$h[0] = s_0$

$h[1] = s_0 + s_1 \cdot p$

$h[r] = s_0 + s_1 \cdot p + s_2 \cdot p^2 --- s_r \cdot p^r$

$h[l-1] = s_0 + s_1 \cdot p --- s_{l-1} \cdot p^{l-1}$

$h[r] - [l-1] = \underbrace{\left(s_l \cdot p^l + s_{l+1} \cdot p^{l+1} --- s_r \cdot p^r\right)}_{p^l}$

$$\left(s_l \cdot p^0 + s_{l+1} \cdot p \; - - - \right)$$

$$\text{Substring}(l - r) = \frac{h[r] - h[l-1]}{p^l}$$

$O(1)$

$$\left(h[R] - h[L-1]\right) \times \text{mod\_inverse}\,(p^{l})$$

log m

0

0

m

$\dfrac{1}{10^6}$ > alphabet size

10^5

a — z

500^{27}

28

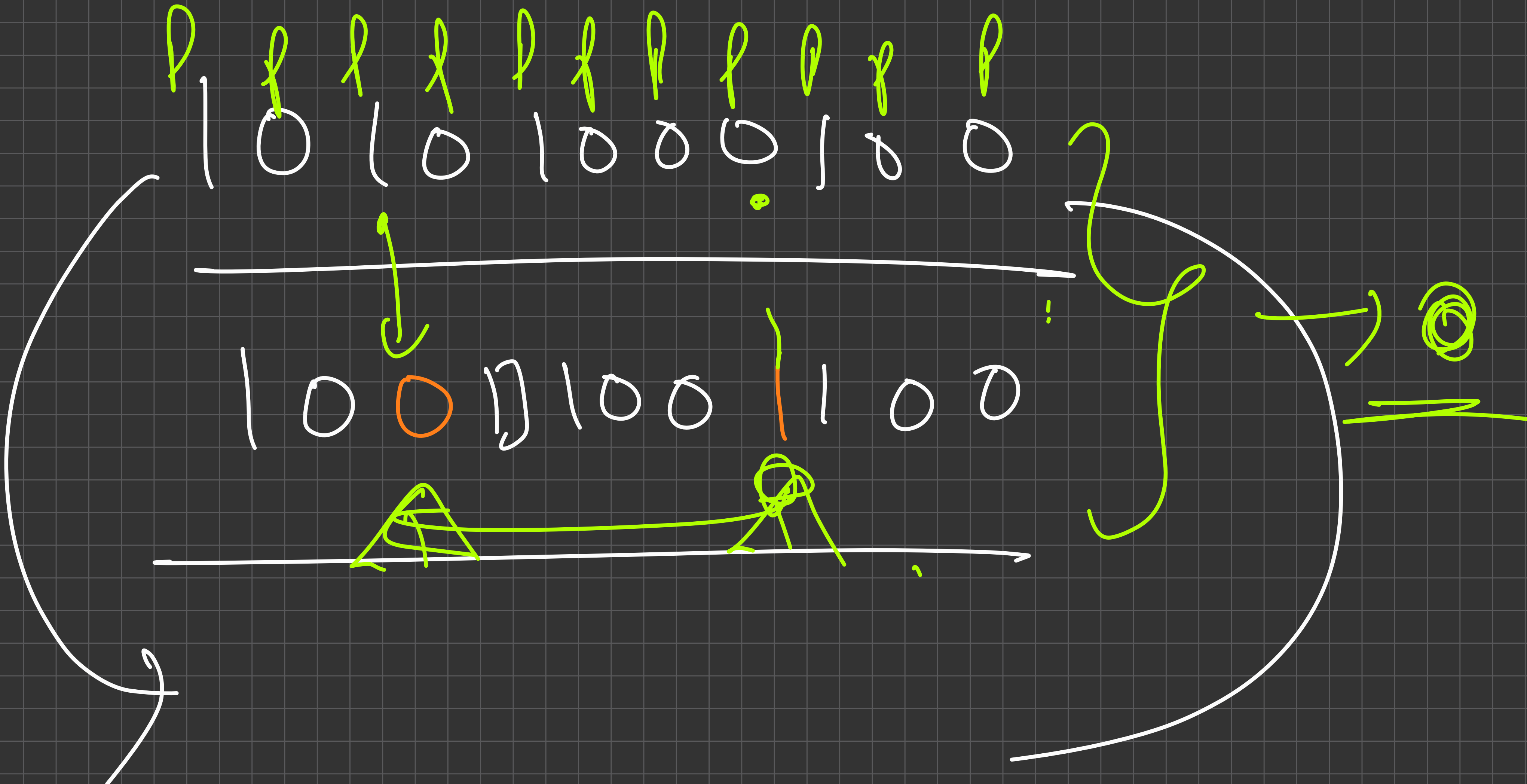
100 29

$$x_0 + x_1 \cdot \boxed{p} + x_2 \cdot p^2 + \dots x_{n-1} \cdot p^{n-1}$$

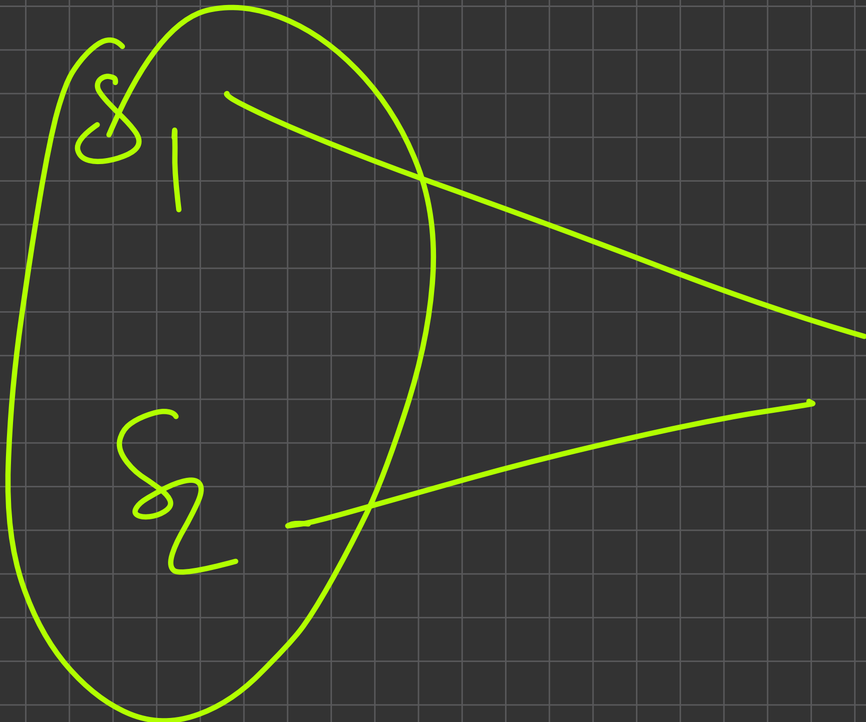
$$1 + 2 \cdot 3 + 15 \cdot 3^2$$

$$> 128$$

$$10^9$$



10101000100



$\underline{\underline{1/m}}$

$\underline{(p, m)}$

(Compare a and f)

1000

1000/m

1/m

$$\frac{1}{m} + \frac{1}{m} + + +$$

$$\underline{\underline{1000/m}}$$

$$\frac{1}{m} = \frac{1}{10^9}$$

$$m = 10^9 + 7$$

$$\downarrow$$

$$\approx 10^9$$

$$1/10^6$$

Codeforces → 10^7 comparisons

$\frac{1}{200}$

WA 1

WA 2

> 1

200 test case

$$\left.\begin{aligned} m_1 &= 10^9+7 \\ m_2 &= 10^9+9 \end{aligned}\right\}$$

$$\left.\begin{matrix} s_1 \\ \\ s_2 \end{matrix}\right\rangle \{h_1, h_2\}$$

$$S_1 = S_2$$

$$\text{hash1}(S_1) == \text{hash2}(S_2)$$

$$s_1 == s_2 \quad \text{if}$$

$$hash1(s_1) == hash1(s_2)$$

$$\&\&$$

$$hash2(s_1) == hash2(s_2)$$

$$\left(\frac{1}{m_1}\right) \times \left(\frac{1}{m_2}\right)$$

$m = 10^{16}$

$m \geq 10^{16}$

$m^2 \leq 10^{18}$

$(a \cdot b) \% m$

$(10^{16} \cdot 10^{16})$

Pitfalls?

- A != B but Hash(A) == Hash(b)
  - Probability = 1 / m
- Comparing 50 such strings, probability of a collision = 50 / m
- Let's look at an example problem to see how it fails
  - Substring comparison problem
- Solution????

Use more primes

Problems

$O(n^2)$ $O(n^3)$ $\boxed{l \quad r}$

$n \le 1000$

- Number of different substrings
- Palindrome queries $O(n+q)$
- Largest string which repeats twice $O(n \log n)$ $O(n^2 \log n) \log n$
- Longest palindromic substring

sus

$O(n)$

$O(n \log n)$

$l \quad r$

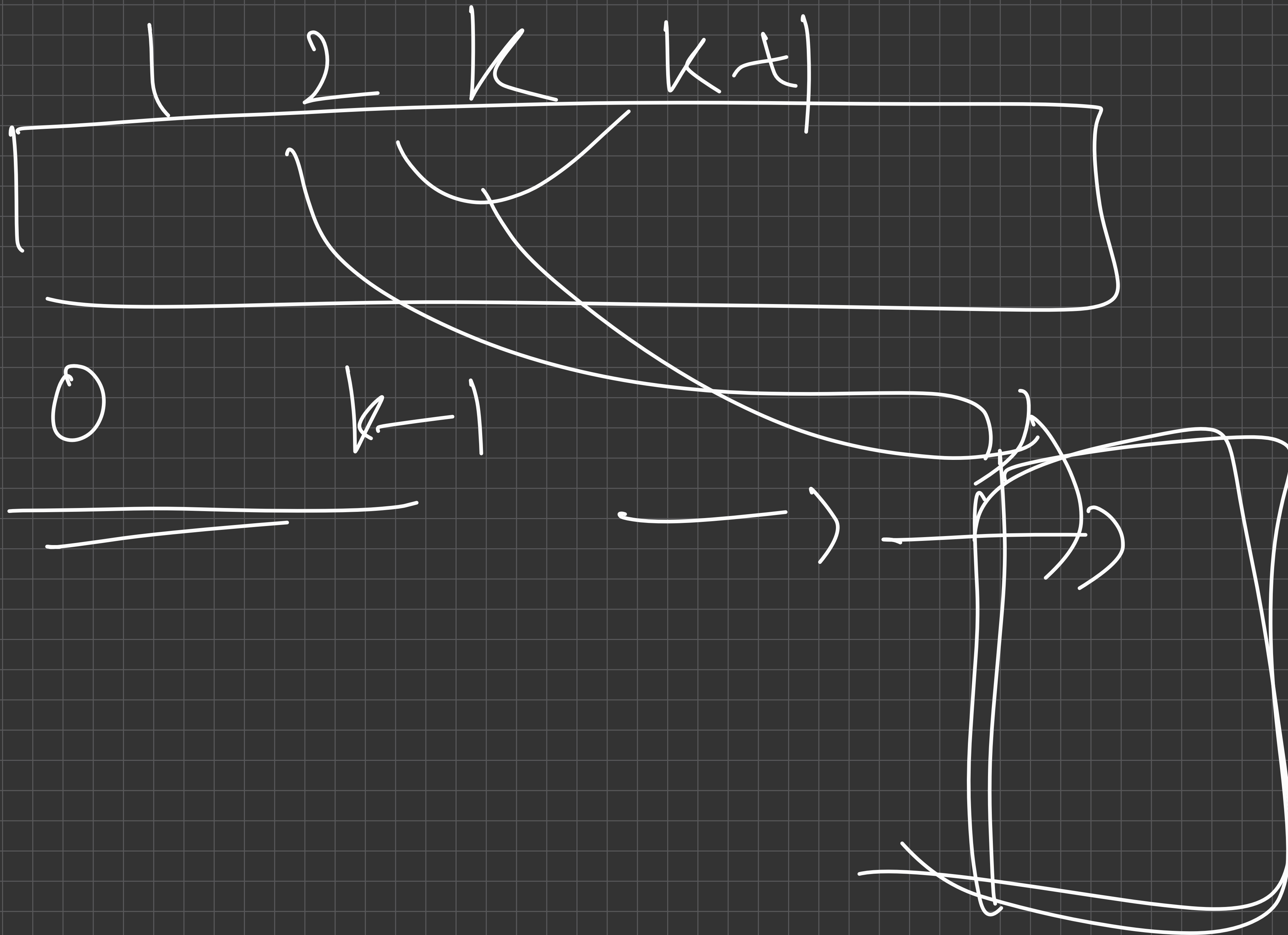
10^5

5

4

3

2


1
2
k
k+1
0
k-1

$O(\log a)$

2

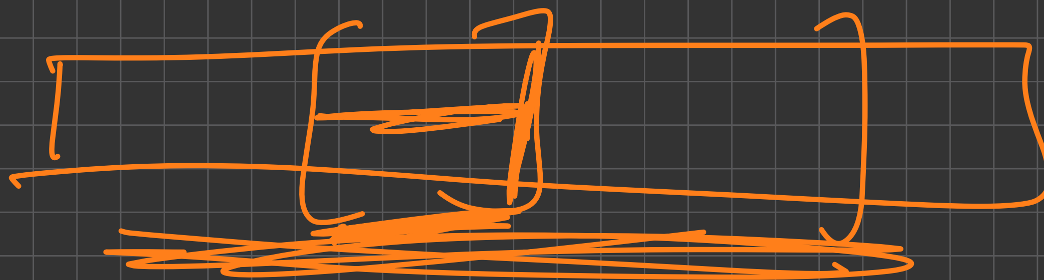
$O(n)$. $O(n\log n)$

2

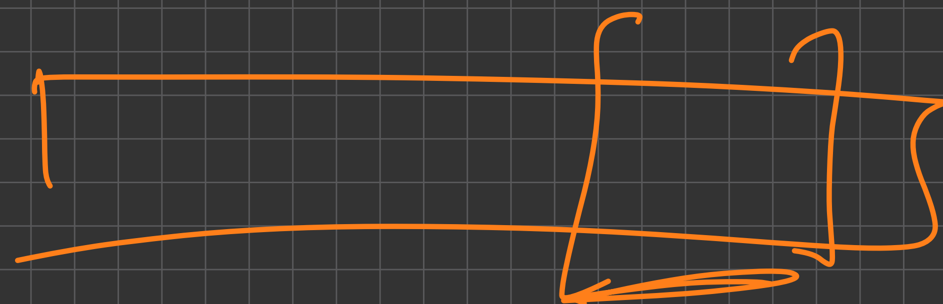
1 2 3 4 5

1 2 3 4 5 6

S



S —



$S \longrightarrow f\,\boxed{adc}\,\boxed{cda}\,g$ $\qquad 0 — 5$

$S\text{-reverse} \longrightarrow g\,\boxed{adc}\,cdaf$ $\qquad 0 — 2$

$2 — 5$

adc adc

adccdg

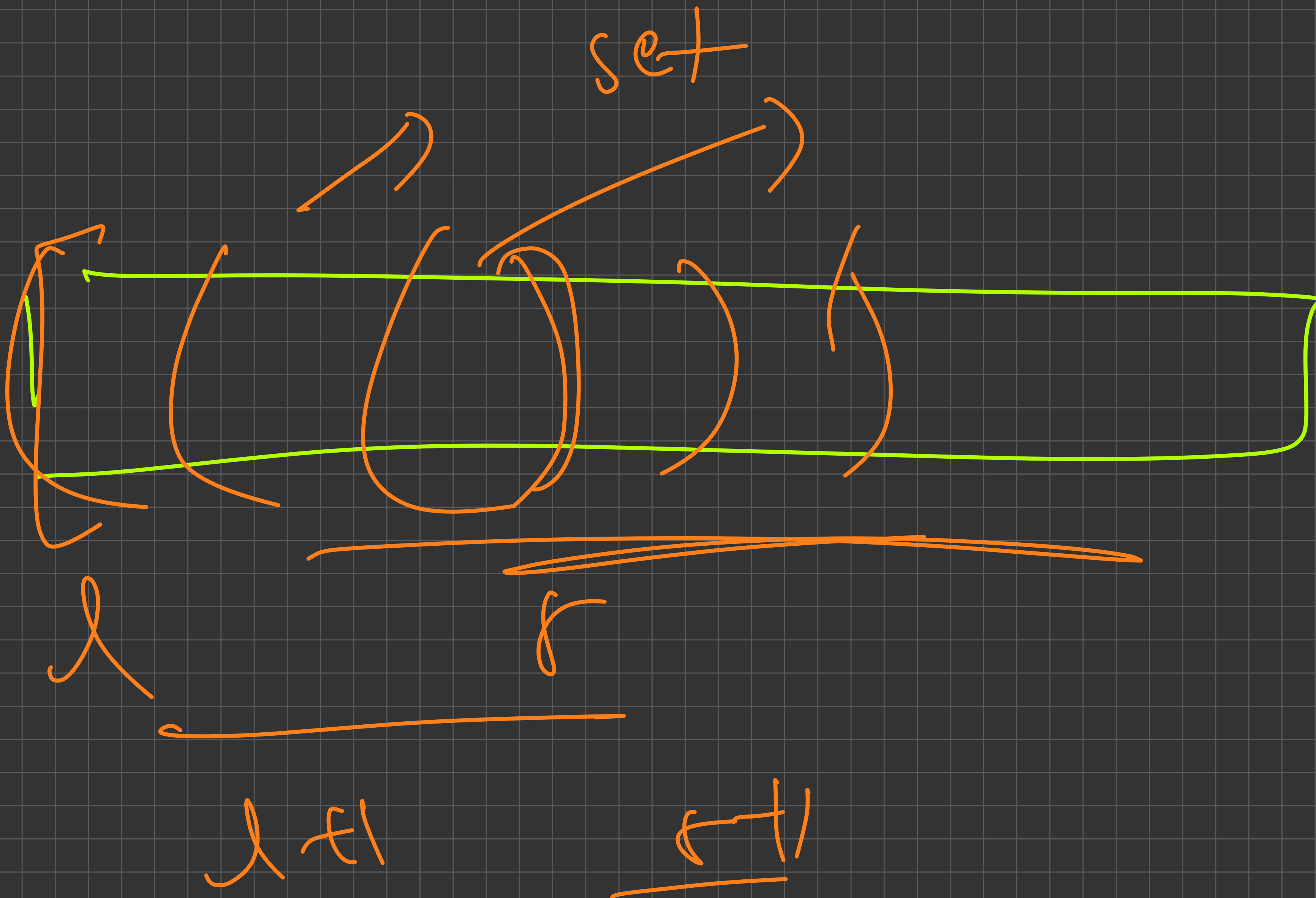
adc != cdg

5 $\longrightarrow$ repeat twice

4 $\longrightarrow$ repeat twice

3 $\longrightarrow$ repeat twice

T	T	T	T	f	f	f
1	2	3	4	5	6	7


set
l
r
l+1
r+1