Introduction to Prime Nos.

Get all primes from 1 to N

Print smallest prime factor for 2 to N

Prime Factorization

Get no. of factors

Mon 30 Oct

Wed 1 Nov

Fri 3 Nov

Prime No.

↳ exactly 2 factors

1 and no. itself

Eg : 1 ✗ 5 ✓ 6 ✗ 7 ✓

1, 5 1, 6, 2, 3 1, 7

Check if a number is prime or not ?

Count no. of factors

Approach 1: i → 1 to N <

```
if (N % i == 0)
    cnt++
```

TC: O(N)

Approach 2: i → 1 to $\sqrt{N}$ <

```
if (N % i == 0) {
    if (i != N/i)
        cnt += 2
    else
        cnt += 1
}
if (cnt == 2)
    print (prime)
else
    print (not prime)
```

TC: $O(\sqrt{N})$

24

i	N/i
1	24/1 = 24
2	24/2 = 12
3	24/3 = 8
4	24/4 = 6

×5

Given a number N, print all prime nos. from 1 to N

	Ans
N = 10	2, 3, 5, 7
N = 20	2, 3, 5, 7, 11, 13, 17, 19

BF : Go to every num from 1 → N and check whether its prime or not

```
for (num = 1 ; num <= N ; num++) {
    for (i -> 1 to √num) {
        if (N % i == 0) {
            if (i != N/i)
                cnt += 2
            else
                cnt += 1
        }
    }
    if (cnt == 2)
        print(num)
}
```

2 + √num + √num

N = 20

num = 1 → i -> 1 to √num

num = 2 → i -> 1 to √2

num = 3 → i -> 1 to √3

num -> 1 to √N

N nums →

TC : O(N√N)

SC : O(1)

1 to $\sqrt{}$

Sieve of Eratosthenes

N = 50

i=

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

all are prime no.

3 → 6, 9 ✓

5 → 10, 15, 20, 25

7 → 14, 21, 28, 35, 42, 49...

4 Not prime

2 → 4 → 8, 12, 16, 20, ...

2 → 6 → 12, 18, 24, 30...

|| → already marked

N = 50

1 → 50

Step 1: Consider 1 → N as prime

bool isprime [N+1] = <true>

isprime [1] = false

arr[51]

idx

0

1

2

.

.

.

50

Step 2: iterate from 2 → N

isPrime [i]

True → i is actually a prime → cancel out multiples

False → i not prime

```
bool isprime[N+1] = {true}

isprime[1] = false
isprime[0] = false

for (i = 2 ; i <= N ; i++) {
    if (isPrime[i] == true) {
        // cancel out multiples
        for(j=2i ; j <= N ; j+=i) {
            isprime[j] = false
        }
    }
}
```

i	j loop iter
2	N/2
3	N/3
4	X
5	N/5
6	X
7	N/7
8 to 10	X
11	N/11

$x \to \frac{1}{x}$

$5 \to \frac{1}{5}$

Total itr $= \frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} + \frac{N}{11} + \dots$

$= N \left(\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{11} + \dots \right)$

↓

sum of prime no.s' reciprocal

$\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{11} + \dots = \log(\log N)$

Total itr $= O(N \log(\log(N)))$

$N = 10^6$ $\quad \log_{10} 10^6 \approx 6$

$\log(\log 10^6) \Rightarrow \log(6) \rightarrow$

i	Multiple of i → mark false
2	2*2 (4), 2*3 (6), 2*4 (8), 2*5 (10), ...
3	3*2 (6), 3*3 (9), 3*4, 3*5, ...
5	5*2, 5*3, 5*4, 5*5, 5*6...

num = a * b

```
bool isprime[N+1] = {true}

isprime[1] = false
isprime[0] = false
for (i = 2 ; i*i <= N ; i++) {        // i <= √N
    if (isPrime[i] == true) {
        // cancel out multiply
        for (j = i*i ; j <= N ; j += i)
            isprime[j] = false
    }
}
```

i	j	itr
$\sqrt{N}$	N	1
$\sqrt{N}+1$	$(\sqrt{N}+1)^2 > N$	0
$\sqrt{N}+2$	$(\sqrt{N}+2)^2$	0

$N + 2\sqrt{N} + 2$

i

i*2 ← 2

i*3 ← 3

i*4 ← 2

i*5 ← 5

i*6 ← 2

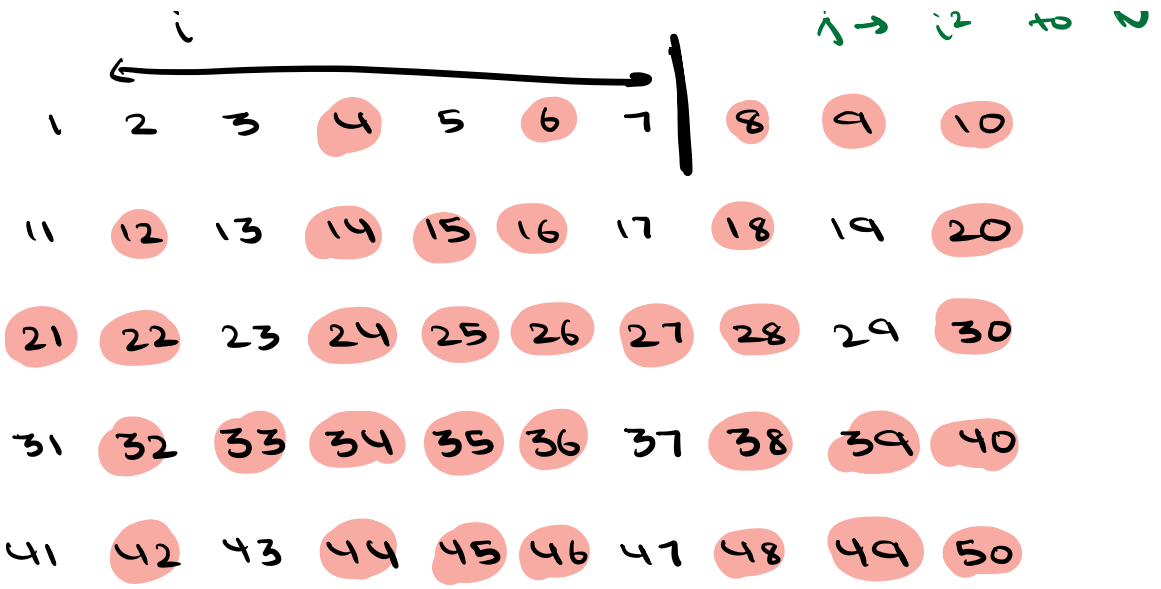
⋮

i*i

N = 50

i → 2 to $\sqrt{50}$

2 to 7



TC : O (N log log N) SC : O(N)

i	j
2	N/2
3	N-9/3
4	x
5	N-25/5

i → 2 to √N

j → i² to N

Given a number N, print smallest prime factor of all nos. from 2 to N

N = 10	2	3	4	5	6	7	8	9	10
spf	2	3	2	5	2	7	2	3	2

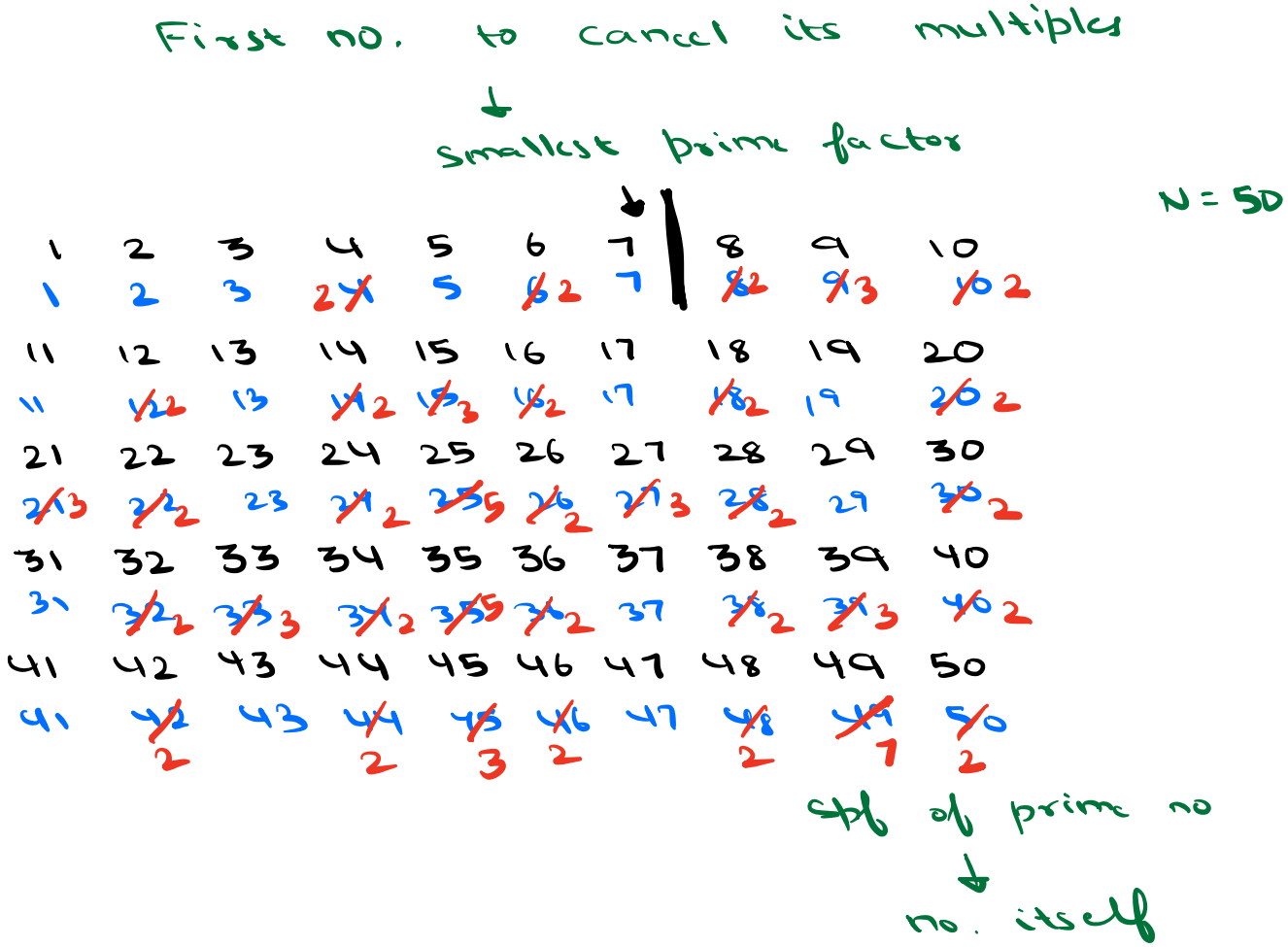
spf → 1 to N

primes → 1 to N

2, 3, 5, 7, 11 . . .

num

Sieve


First no. to cancel its multiples
smallest prime factor
N = 50
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
spf of prime no
no. itself

```
int spf[N+1]                                   1 -> N
                                               idx
// every no. is claiming to be prime           0
for (i = 2 ; i <= N ; i++)                     1
    spf[i] = i                                 :
                                               :
                                               N
                  i <= √N
for (i = 2 ; i*i <= N ; i++) {
    if (spf[i] == i) {
        for (j = i*i ; j <= N ; j += i) {
            if (spf[j] == j) {
                spf[j] = i
            }
        }
    }
}
```

i	j
2	4, 6, 8, 10, 12.
3	9, 12

TC: $O(N \log(\log N))$

SC: $O(N)$

Prime Factorization

↓

process of representing a no. as product of prime factors alone

$20 = 10 \times 2$ ✗

$= 5 \times 4$ ✗

$20 = 2 \times 2 \times 5$

$n = 48$

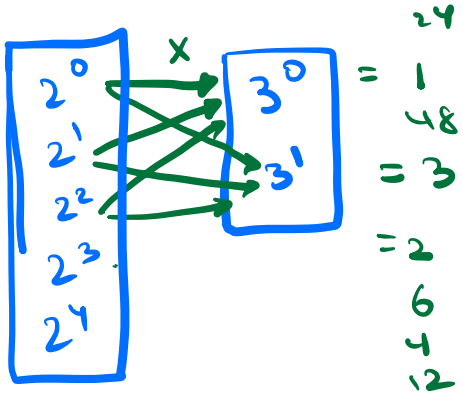
2	48
2	24
2	12
2	6
3	3
	1

Cnt of factors of 48 = 10

1, 2, 3, 4, 48, 24, 16, 12, 6, 8

$48 = 2 \times 2 \times 2 \times 2 \times 3$

$48 = 2^4 \times 3^1$



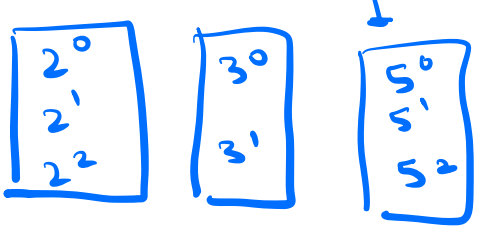
$48 \rightarrow 5 \times 2$

$= 10$ factors

$n = 300$

2	300
2	150
3	75
5	25
5	5
	1

$300 = 2^2 \times 3^1 \times 5^2$



$3 \times 2 \times 3$

300 → 18 factors

Assume n's prime factorization

$$n = p_1^{a_1} \; p_2^{a_2} \; p_3^{a_3} \; \cdots \; p_z^{a_z}$$

$[0, a_1]$ $[0, a_2]$ $[0, a_3]$ ... $[0, a_2]$

a_1+1 a_2+1 a_3+1 ... a_2+1

Total no. of factors of $n = (a_1+1) \times (a_2+1) \times (a_3+1) \times \ldots \ldots (a_2+1)$

$40 = 2^3 \times 5^1$

No. of factors $= (3+1) \times (1+1)$

$= 4 \times 2$

$= 8$

ans = 1

ans = 1 * (3+1)

ans = 4

ans = 4 * (1+1)

= 8

2	40
2	20
2	10
5	5
	1

No. of factors 1 → N

SPF 1 → N

```
int count (int N, int spf[ ]) {
    ans = 1
    while (N > 1) {
        int x = spf[N]
        int cnt = 0
        while (N % x == 0) {
            cnt++
            N = N / x
        }
        ans *= (cnt + 1)
    }
}
```

how many times can I divide by spf x

N → N/2 → N/4 → N/8 → ... → 1

TC : $O(\log_2 N)$

Count factors of all nos. from 1 → N

	1	2	3	4	5	6	7	8	9	10
	1	1,2	1,3	1,2,4	1,5		1,7		1,3,9	
N=10	1	2	3	4	5	6	7	8	9	10
cnt	1	2	2	3	2	4	2	4	3	4

① Build spf[N+1] → N log log N

② for (num = 1 ; num ≤ N ; num++) {
    count (num, spf[ ])
}
→ N log N

TC : O(N log N) SC : O(N)

Approach 2 →

N = 20 1 → 20

1	2	3	4	5	6	7	8	9	10
1	2	2	3	2	~~4~~	2	~~4~~	~~3~~	~~4~~

11	12	13	14	15	16	17	18	19	20
2	~~3~~ 4 ~~5~~ 6	2	~~3~~ 4	~~2~~ ~~3~~ 4	~~4~~ 5	2	~~3~~ ~~4~~ ~~5~~ 6	2	~~3~~ ~~4~~ ~~5~~ 6

i = 2 → N/2

j = 2i

```
int col[N+1] = {2}
col[1] = 1
```

i ≤ N/2

```
for (i = 2 ; i ≤ N ; i++) {
    for (j = 2*i ; j ≤ N ; j+=i) {
        col[j] += 1
    }
}
```

SC : O(N)

TC : O(N log N)

i	j (2*i)	
N/2	N	→ 1
N/2 + 1	→ N	→ 0

i	$i \cdot \frac{1}{i}$
2	$2 \cdot \frac{1}{2}$
3	$2 \cdot \frac{1}{3}$
4	$2 \cdot \frac{1}{4}$
5	$2 \cdot \frac{1}{5}$
6	$2 \cdot \frac{1}{6}$
7	$2 \cdot \frac{1}{7}$
⋮	⋮

total is

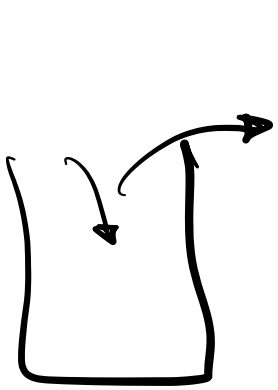
$$= 2\cdot\frac{1}{2} + 2\cdot\frac{1}{3} + 2\cdot\frac{1}{4} + 2\cdot\frac{1}{5} + 2\cdot\frac{1}{6} + 2\cdot\frac{1}{7} + \cdots$$

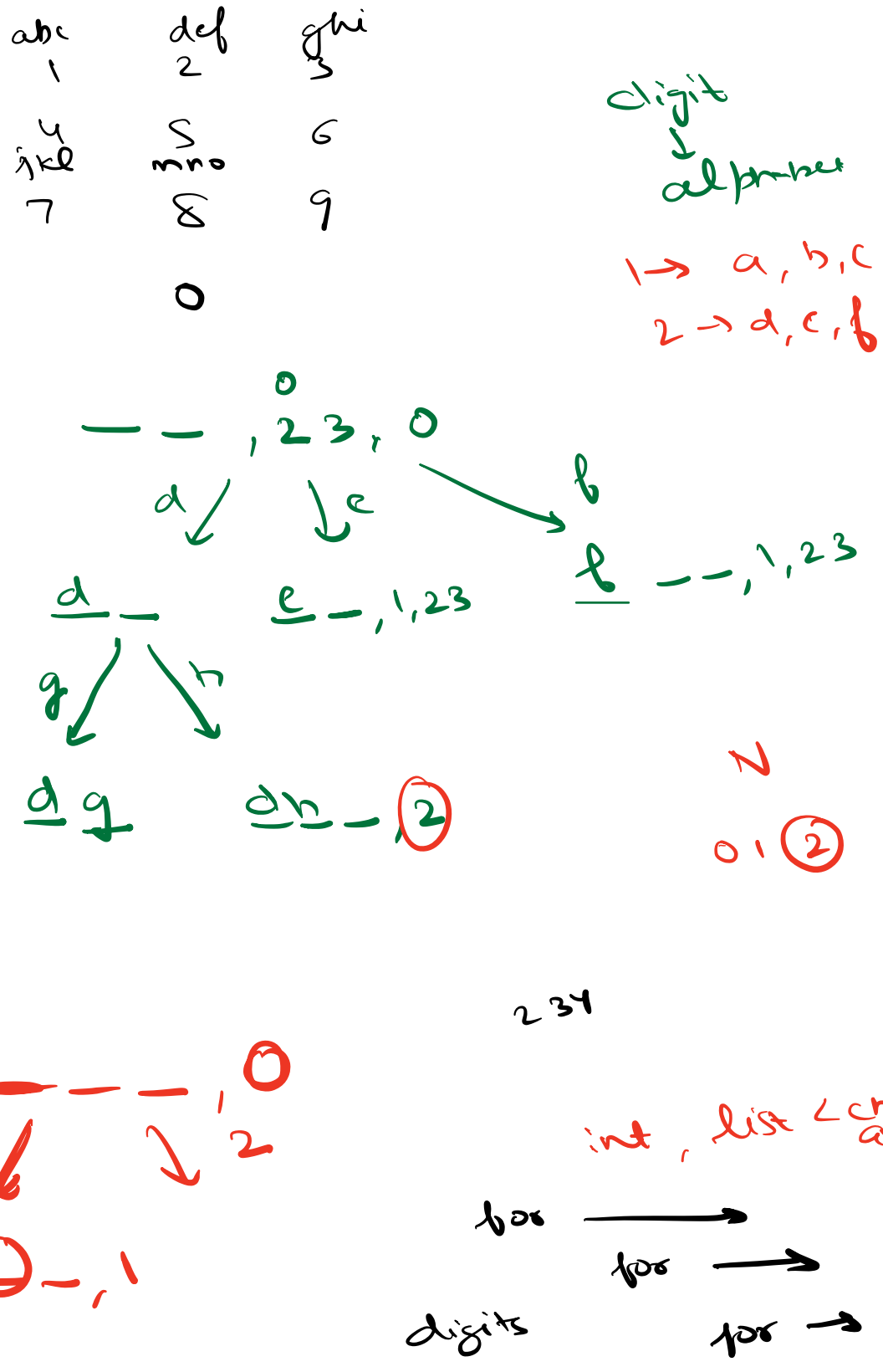
$$= 2\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \cdots\right)$$

$$= 2 \log 2$$

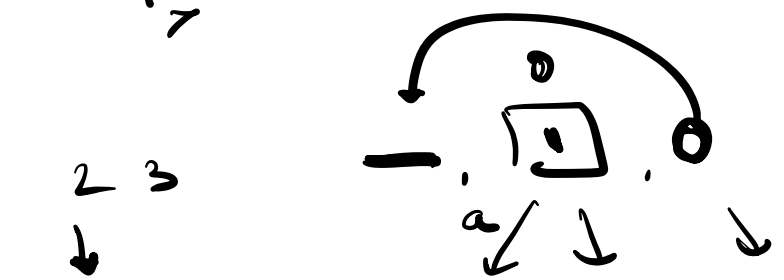
$$\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \cdots \quad 2\sum_{i=2} \frac{1}{i}$$

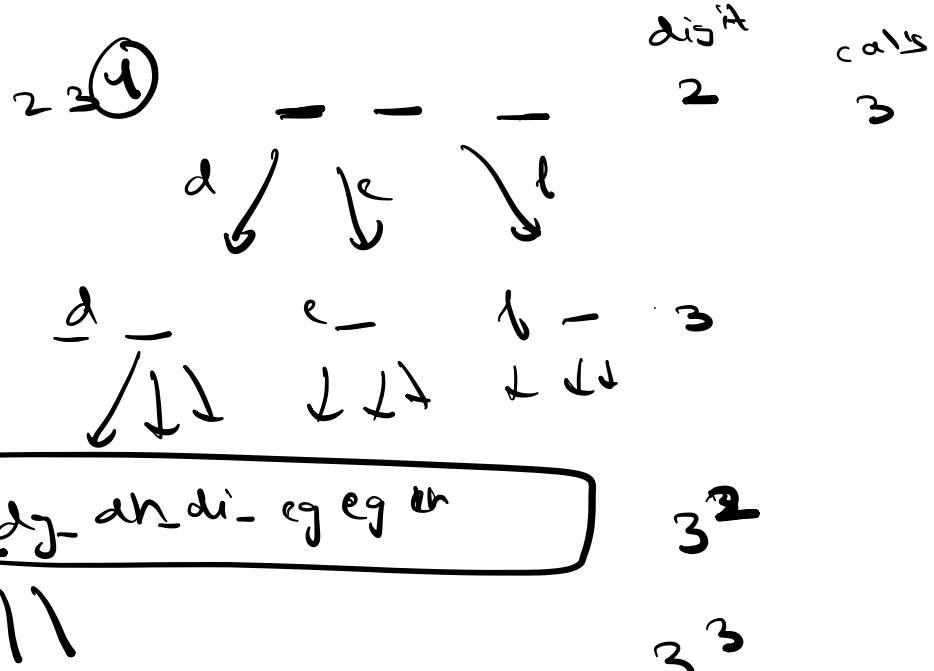
$$= \log 4$$

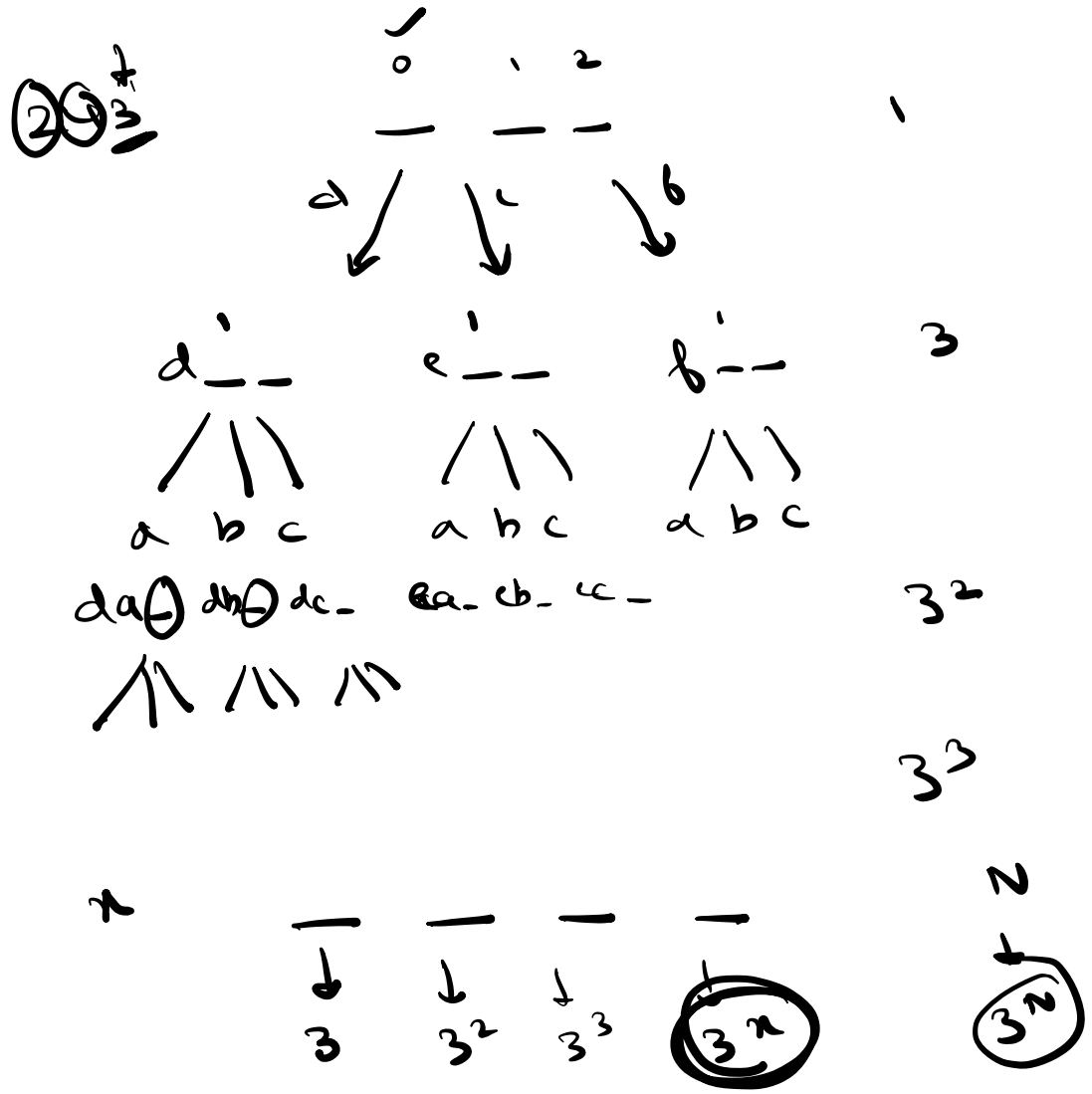



abc def ghi
1 2 3
4
jkl 5 mno 6
7 8 9
0
digit
alphabet
1 → a, b, c
2 → d, e, f
0
_ _ , 2 3 , 0
d e f
1, 23 d _ e _ , 1, 23 f _ _ , 1, 23
g h
2, 23, dg dh _ , 2
N
0 , 2
2 34
_ _ _ , 0
1 2
1 0 _ , 1
int , list < ch
for
for
digits for
list < string > ans

```
void genwords(int dig[], int N, int idx, mp) {
    if (idx == N) {
        ans.add(str)
        return
    }
    digit = dig[idx]
    for (char in mp[digit])
        curr = char
        genwords(dig, N, curr, idx+1, mp)
        curr.pop_back()
}
```






0 1 2
a b c
d e f
a b c a b c a b c
3
3²
3³
n
3
3²
3³
3ⁿ
3ⁿ