Precedence

+ , − → 1

/ , * → 2

Infix Conversion

/ ≤ *

lopt = *

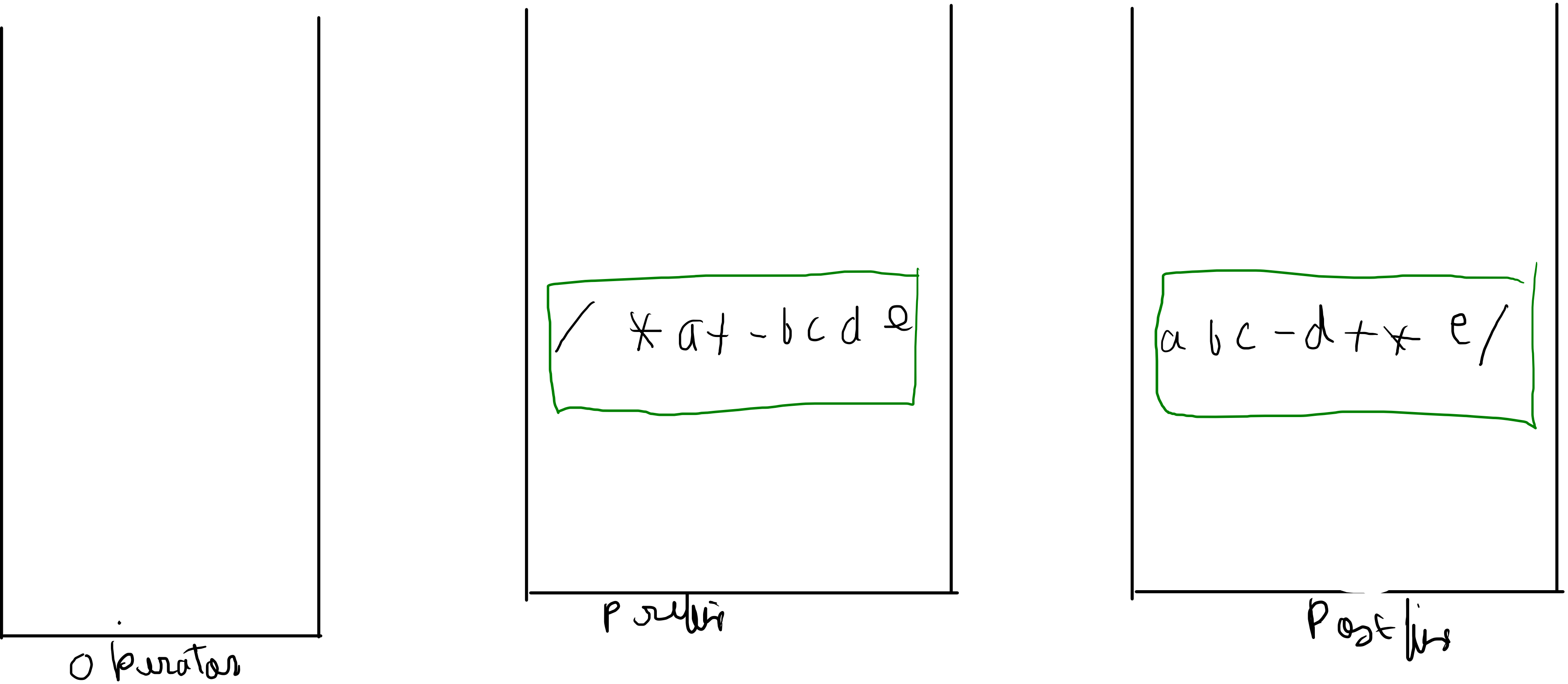
Post → val2 = val1 = .

bc -d+

a

str = a * (b - c + d) / e

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑



Postfix Eval & Convert

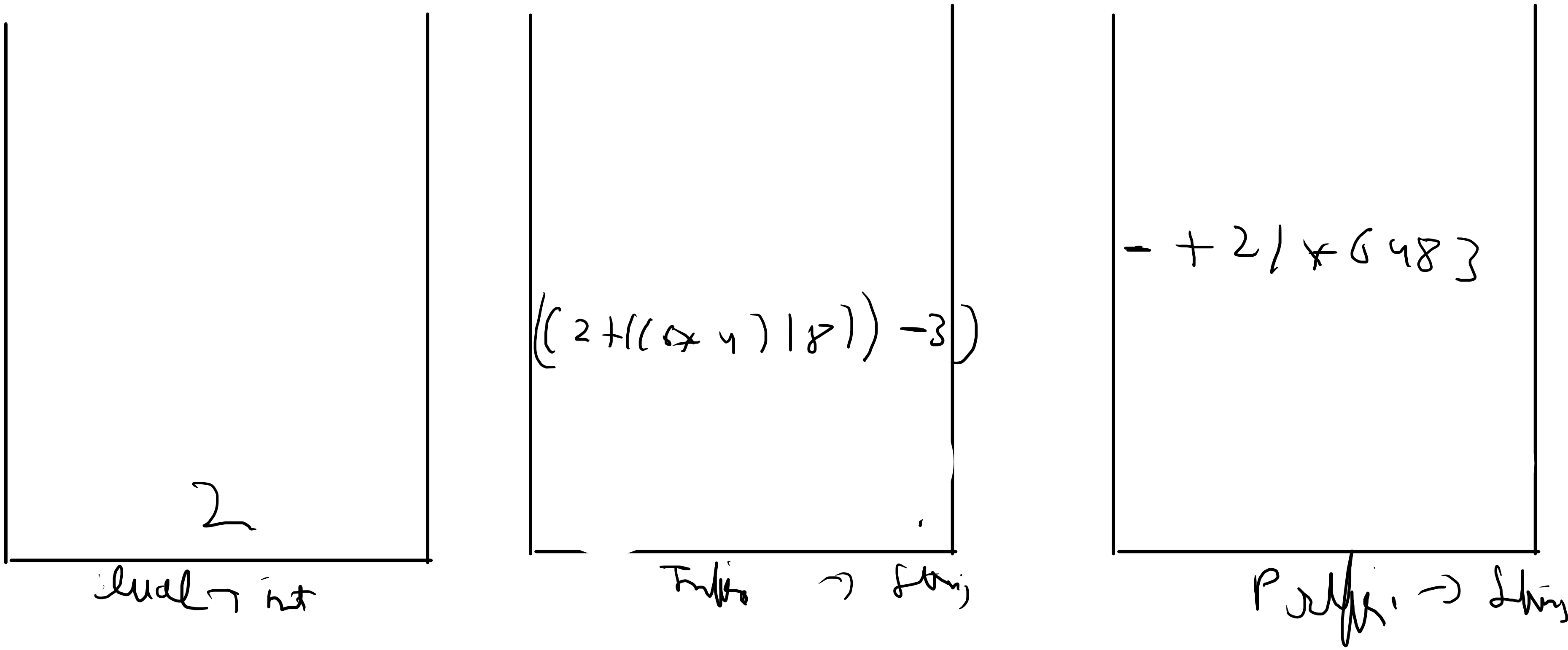
opt = −

v_2 = 3

v_1 = (2 +

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

Str = 2 6 4 * 8 / + 3 −



Prefix → start from last & same as Postfix

↓

$- + 2 / * 6 4 8 3$

Stable & Unstable

Same element coming in order as I/P

I/P ⇒ $5' \quad -2 \quad 4' \quad 5'' \quad 3 \quad 4'' \quad 5'''$

O/P ⇒ $-2 \quad 3 \quad 4' \quad 4'' \quad 5' \quad 5'' \quad 5'''$ → stable

$-2 \quad 3 \quad 4' \quad 4'' \quad 5'' \quad 5''' \quad 5'$ → unstable

Rules

① if (ch == digit)

→ Push

② if (ch = +, -, *, /)

2 + 6 ~~8~~

+ 26 → Pre

26+ → Post

① if(ch == ( ) → Push opt Stack

② if (ch = a-z / A-Z | 0-9)

→ Push

Postfix & Prefix Stack

③ if (ch = +, -, /, *)

→ while(opt.size() > 0 && opt.peek != (&&

Priority (ch) ≤ Priority (opt.peek)) {

opt = opt.peek()

Prefix

val2 = Prefix.pop()

val1 = Prefix pop

$\boxed{\text{ans} = \text{opt} + \text{val1} + \text{val2}}$ → Push

Postfix

val2 = Postfix.pop()

val1 = - - -

ans = val1 + val2 + opt

④ if (ch ==))

→ while (( )