

CS & IT ENGINEERING

Compiler Design

Intermediate code and code optimization



Lecture No. 4



By- DEVA Sir

Data Flow Analysis

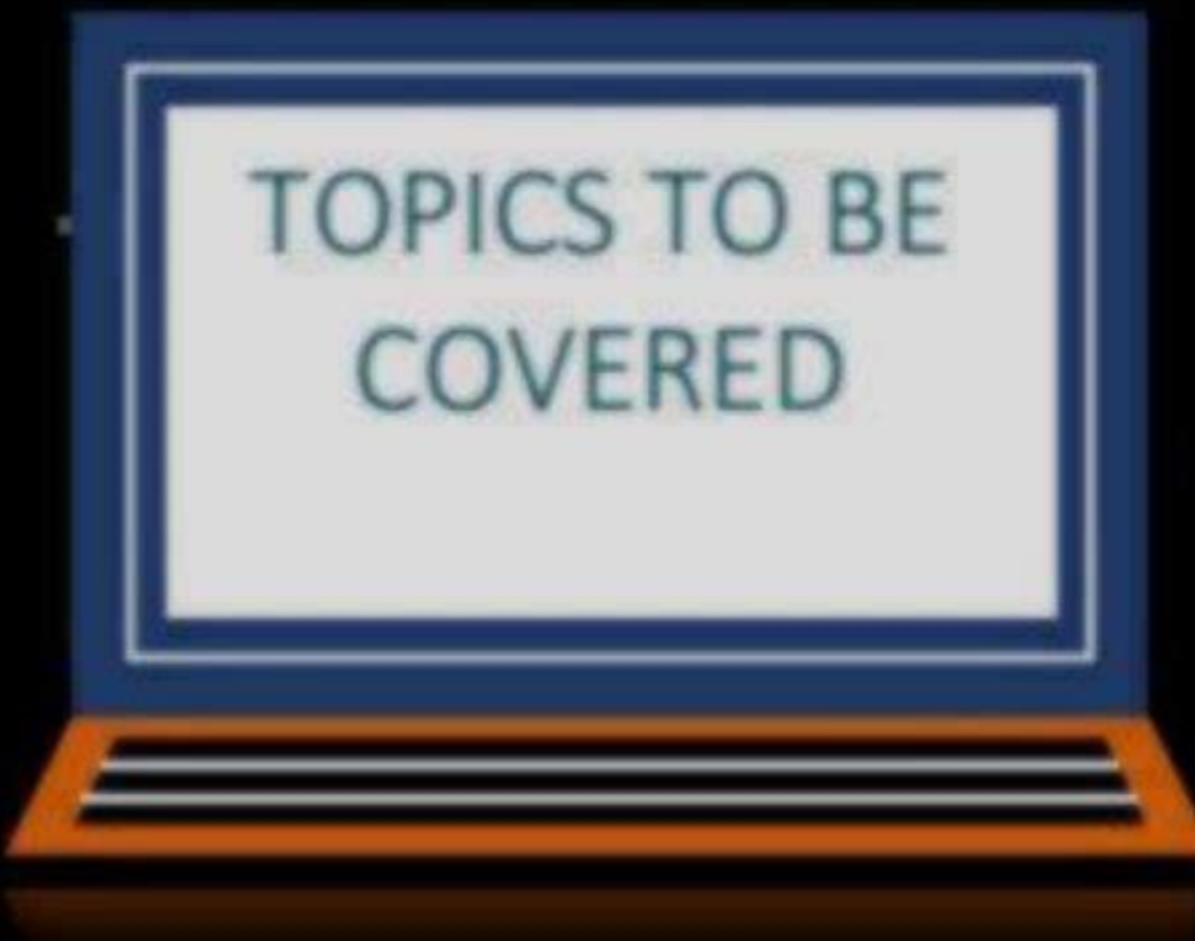


1) Live Variable Analysis

- ↳ What is live variable?
- ↳ GEN, KILL, IN and OUT sets
- ↳ Live variable analysis

2) Reaching Definitions Analysis

- ↳ USE-defs chains
- ↳ Def-uses chains



What is Live variable ?

1. $x = a + b$

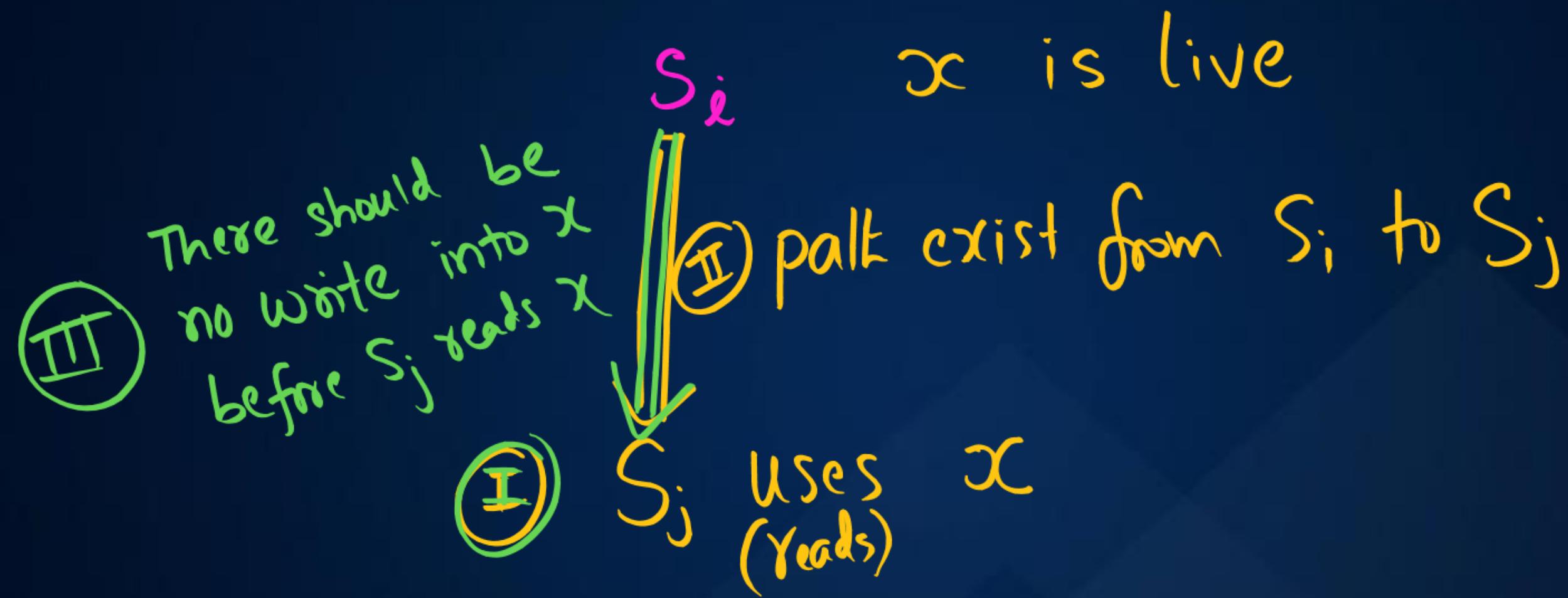
2. $y = c * d$

3. $z = \textcircled{x} - c$

4. $w = y * z$

IS x live at statement 2 ?

2 $\xrightarrow[\text{fall}]{\text{nowrite in } x}$ 3
reads x



x is live variable at statement S_i

iff

(Read)

(P alk)

(No write)

I) Some statement S_j reads x

II) Path exist from S_i to S_j

III) No write into x inbetween S_i and S_j
before reading x .

Q1)

Find live variables at statement 1

$$1. \quad x = a + b$$

a
l → l
l → 2 → 3 → 4

statement 1

b
l → l
l ↓ 2

c
x
y
z

$$2. \quad y = x * c$$

$$3. \quad z = x + y$$

$$4. \quad a = b * z$$



Q2) Find live variables at statement 2.

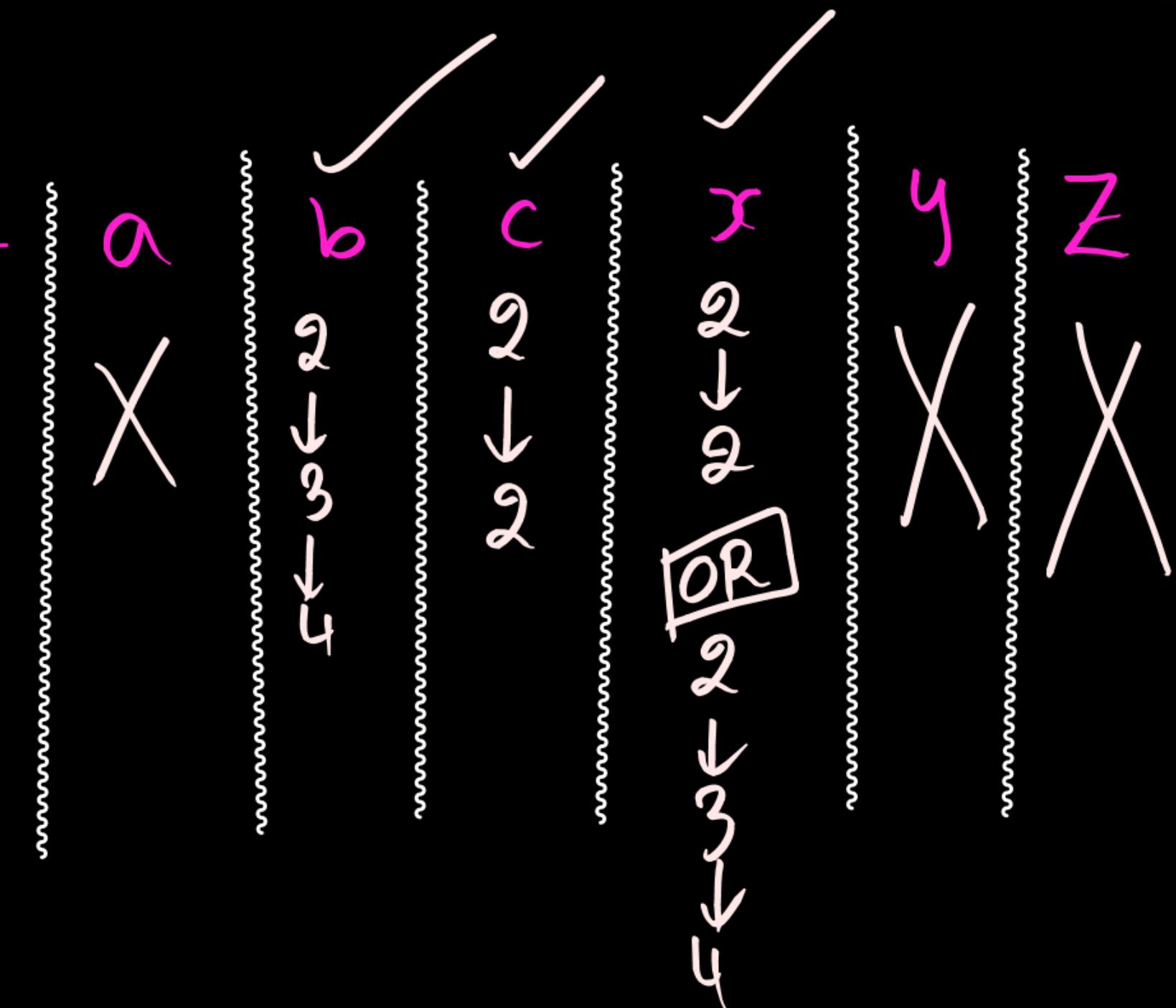


$$1. x = a + b$$

$$2. y = \cancel{x} * c$$

$$3. z = \cancel{x} + y$$

$$4. a = \cancel{b} * z$$



Q3) Find live variables at statement 3.

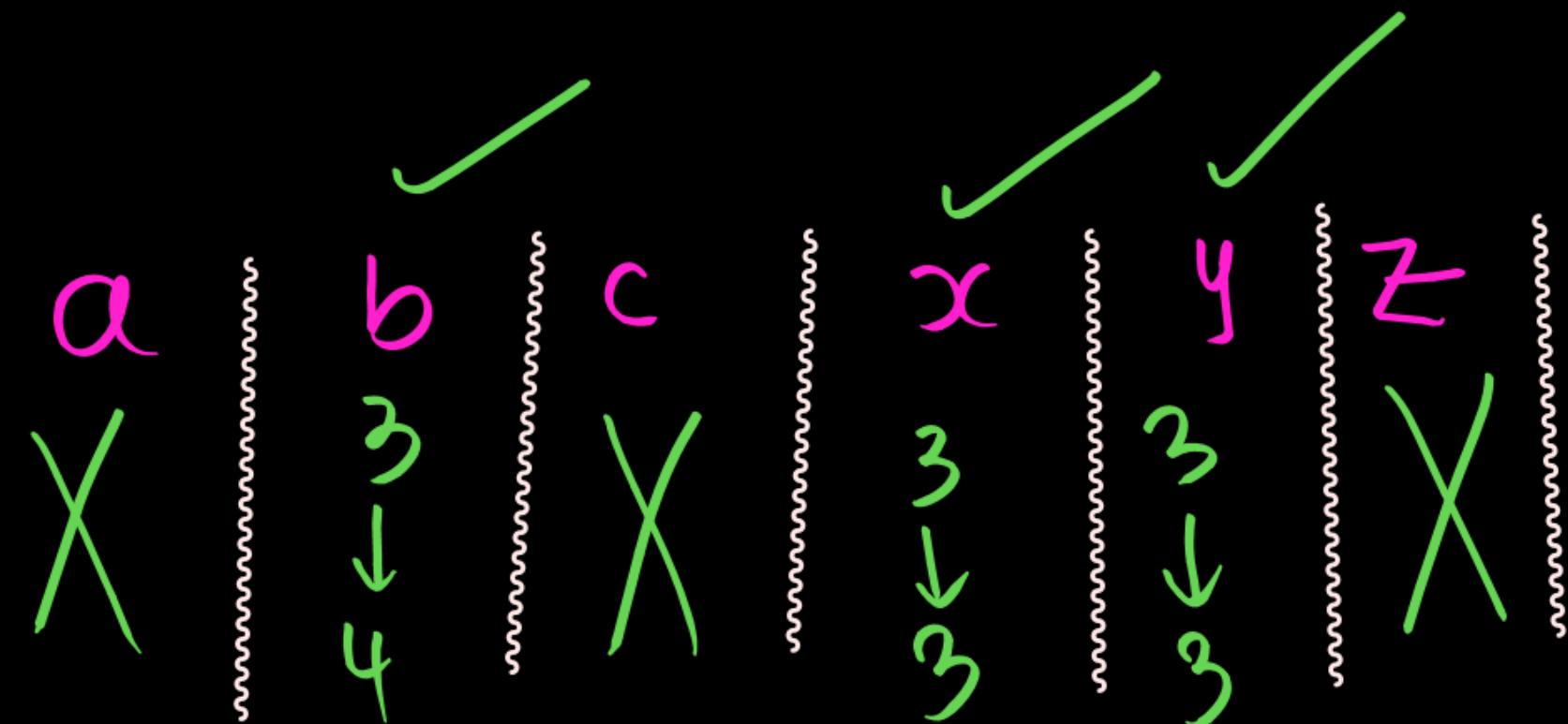


1. $x = a + b$

2. $y = x * c$

3. $z = \cancel{x} + \cancel{y}$

4. $a = \cancel{b} * z$



Q4) Find live variables at statement 4.

1. $x = a + b$

2. $y = x * c$

3. $z = x + y$

4. $a = b * z$ ← b, z are live

φ is live at S_i

Some S_j reading φ

no write into φ

GEN Set and KILL Set for Basic Block :



$$GEN_k = \{a, b, c, y\}$$

= Set of variables
in BB where
every variable

used at some statement in BB
but no write into it before read

$x = a + b$
 $c = x - c$
 $z = x * y$

BB ↵

$$KILL_k = \{x, c, z\}$$

= Set of variables
which are
defined in BB.

$$= \{v \mid v \text{ is defined in BB}\}$$

what happens after read

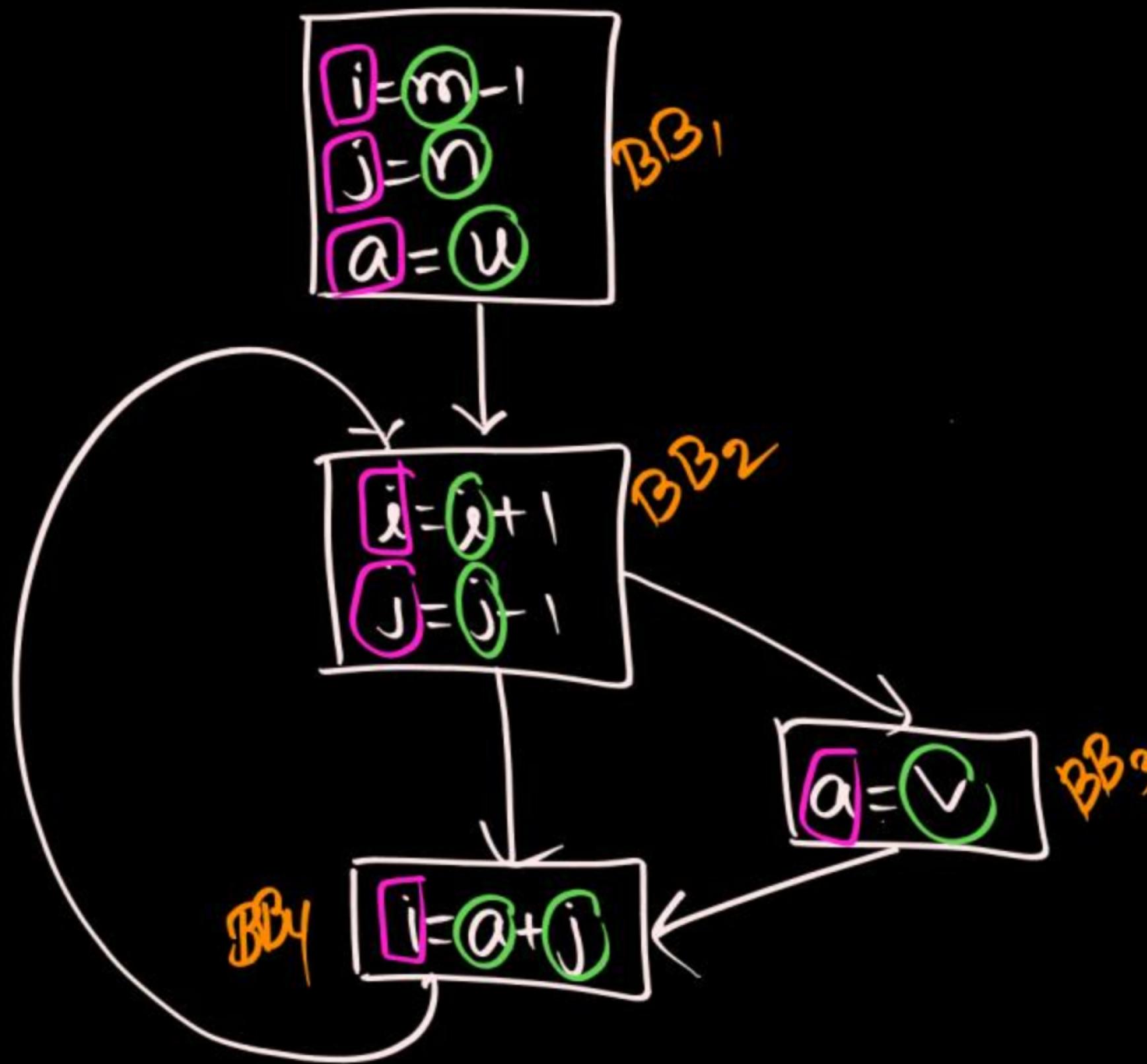
$C = x - C$ reads

this happens final

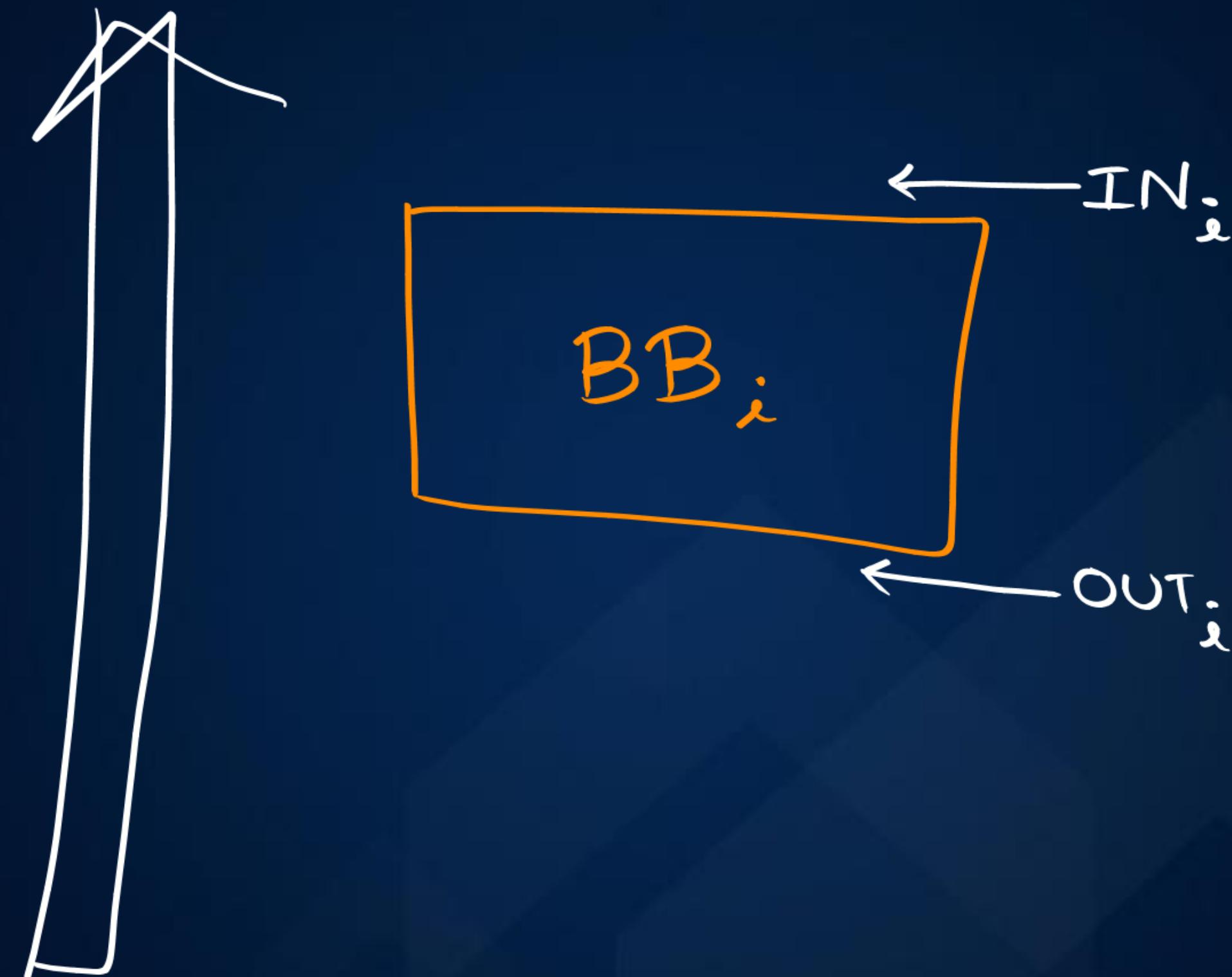
IS C live?

C is live

Find GEN^{set} and KILL Set for Every Basic Block.



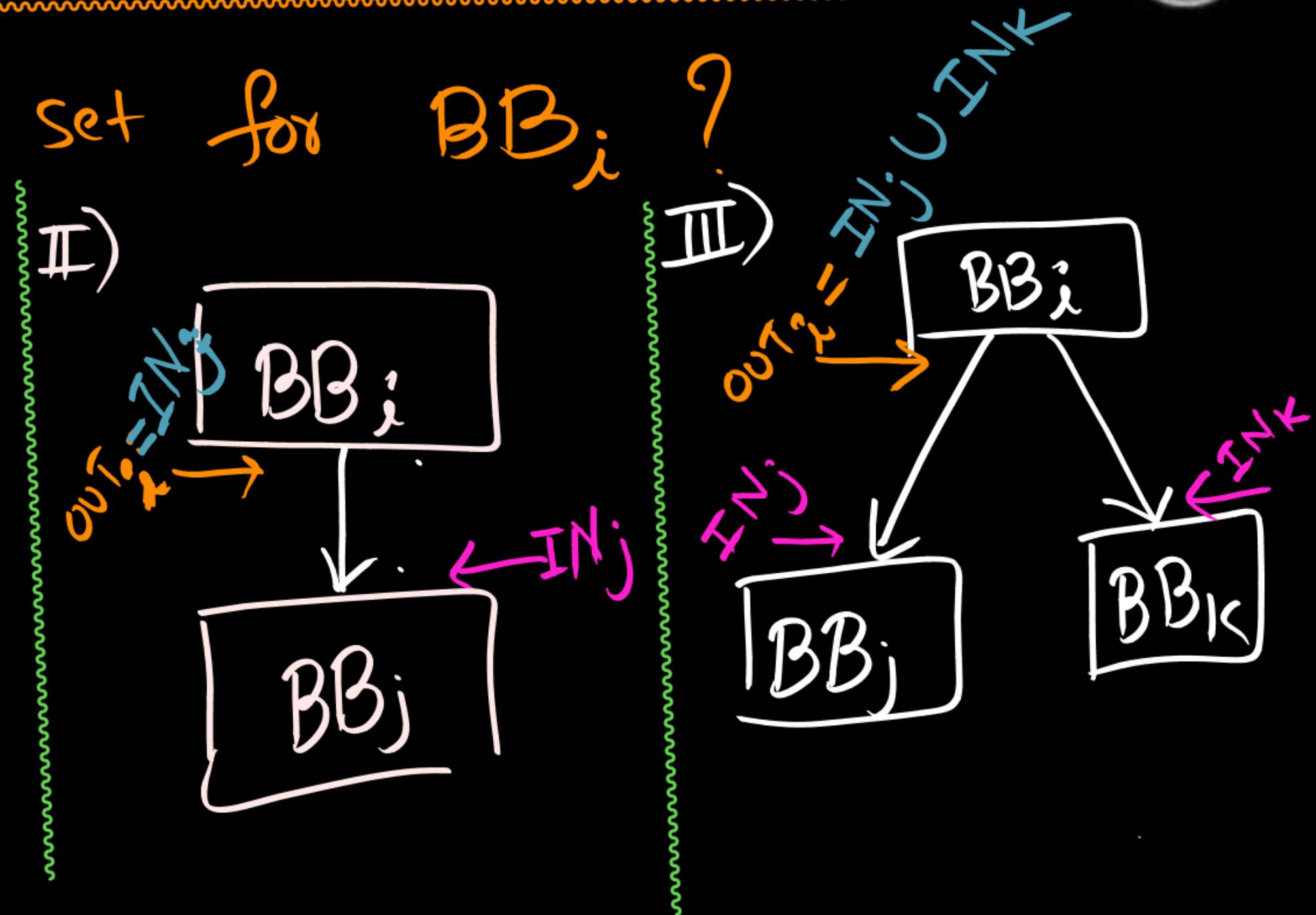
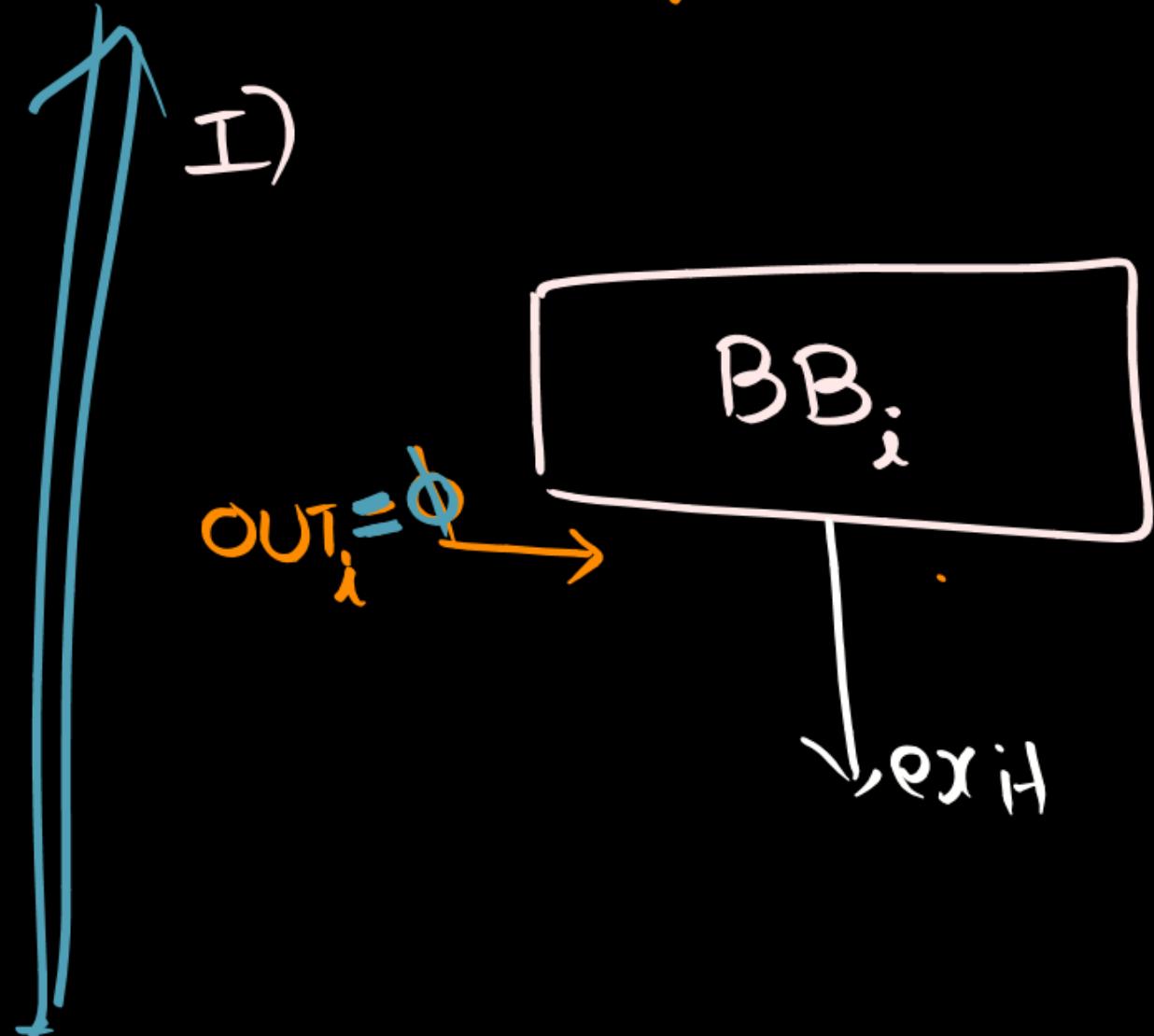
	GEN_{set}	KILL Set
BB_1	$\{m, n, u\}$	$\{i, j, a\}$
BB_2	$\{i, j\}$	$\{i, j\}$
BB_3	$\{v\}$	$\{a\}$
BB_4	$\{a, j\}$	$\{i\}$



IN Set and OUT Set for Basic Blocks :



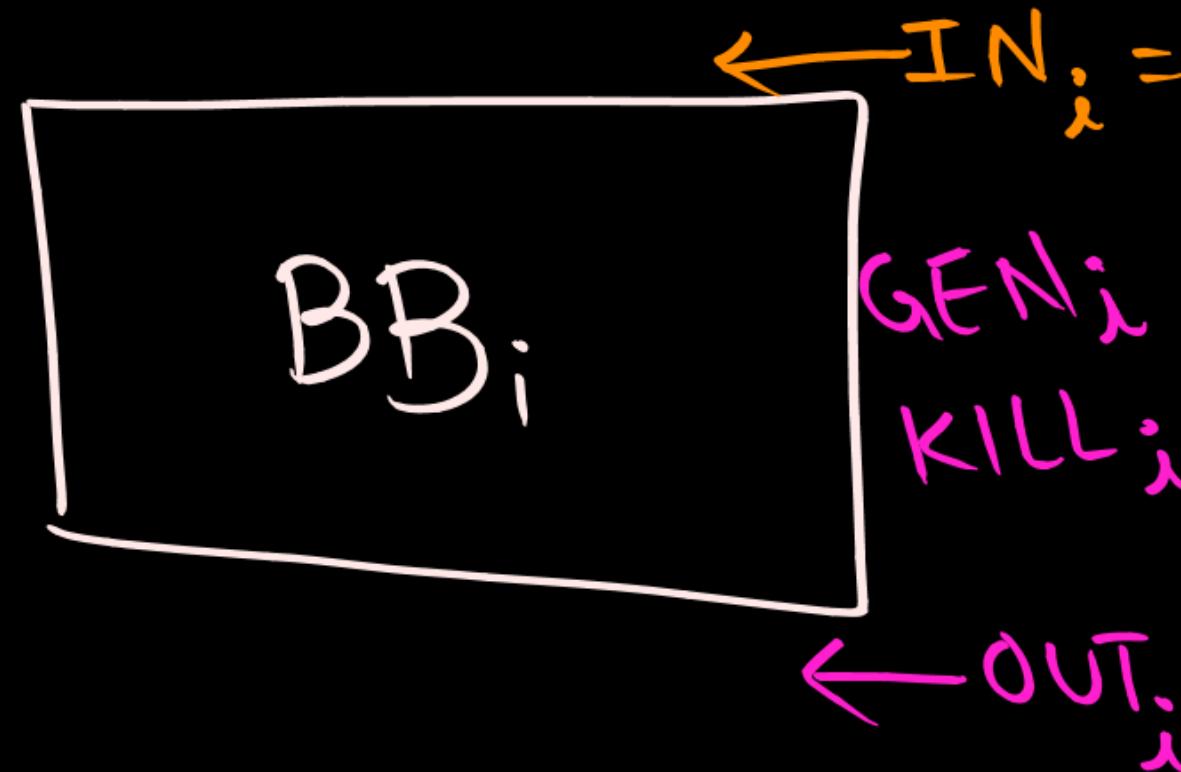
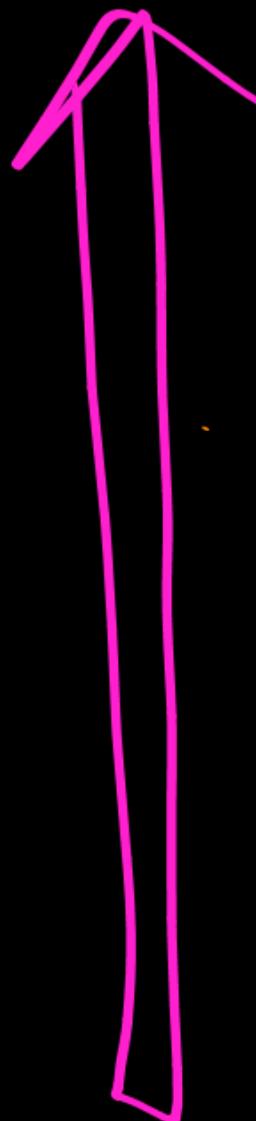
How to Compute OUT set for BB_i ?

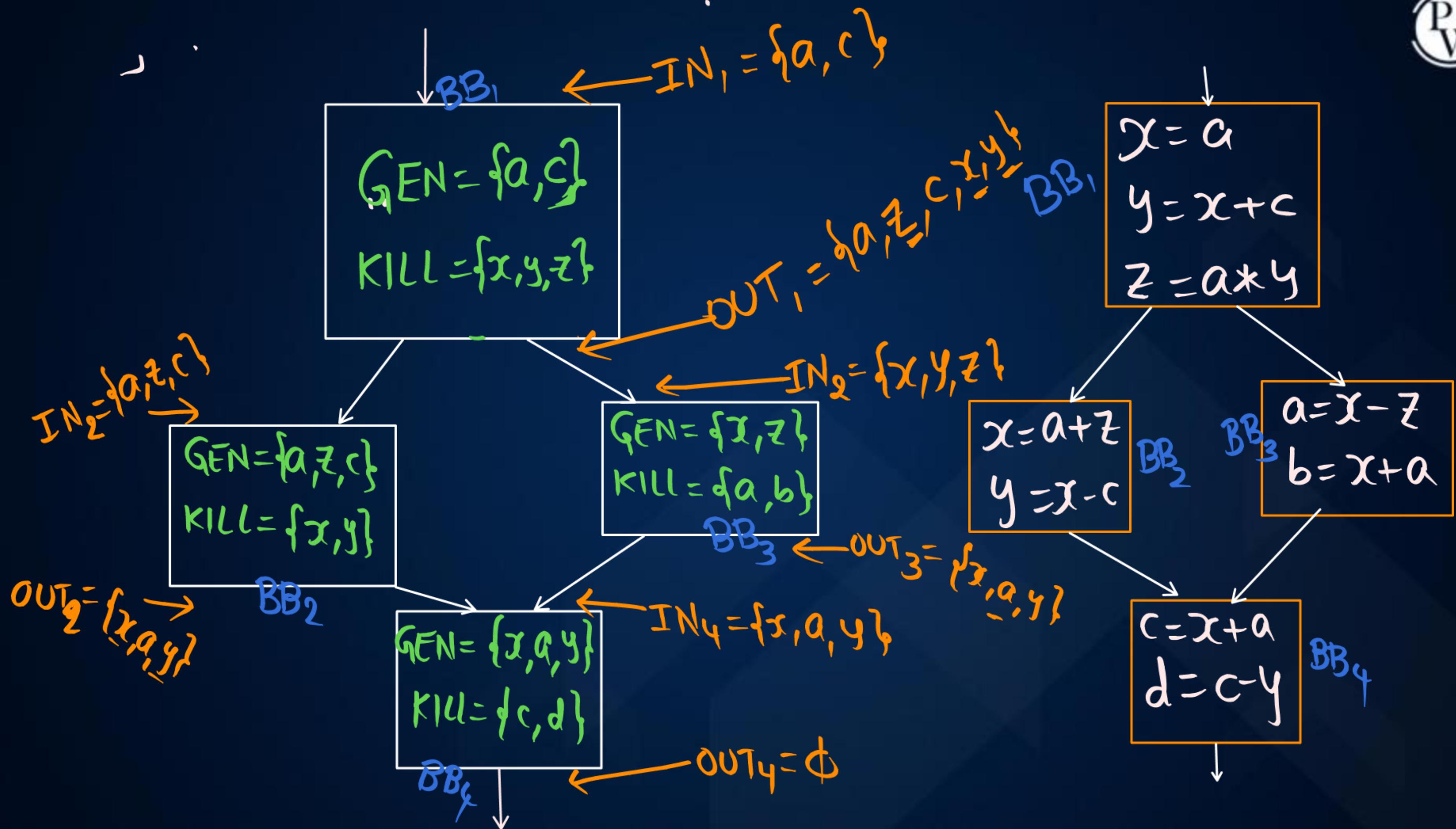


IN Set and OUT Set for Basic Blocks :



How to Compute IN Set for BB_i ?





1 $a = a + 1$
2 $b = a + c$
3 $c = b * 2$

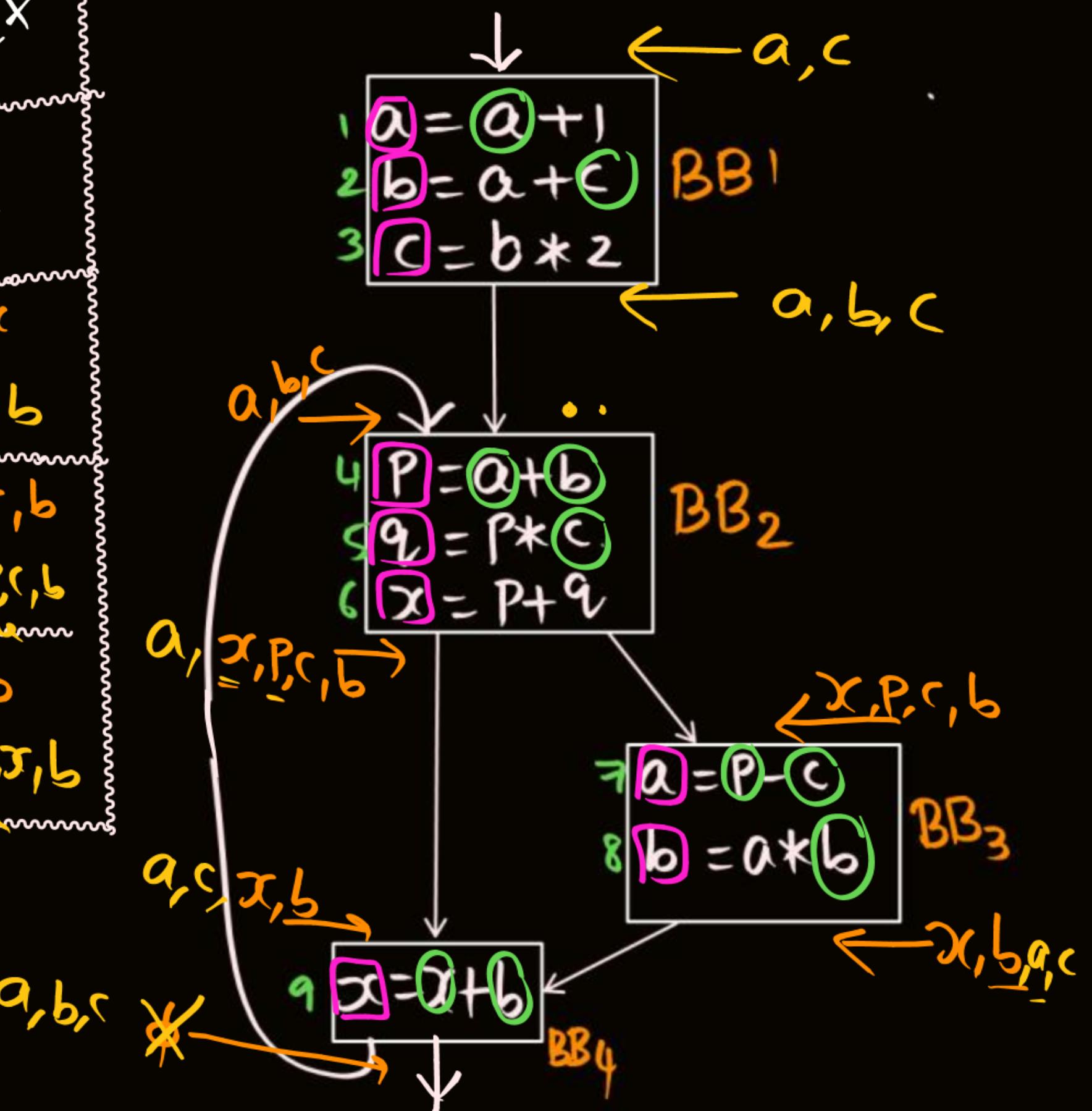
4 $P = a + b$
5 $q = P * c$.
6 $x = P + q$

7 $a = P - c$
8 $b = a * b$

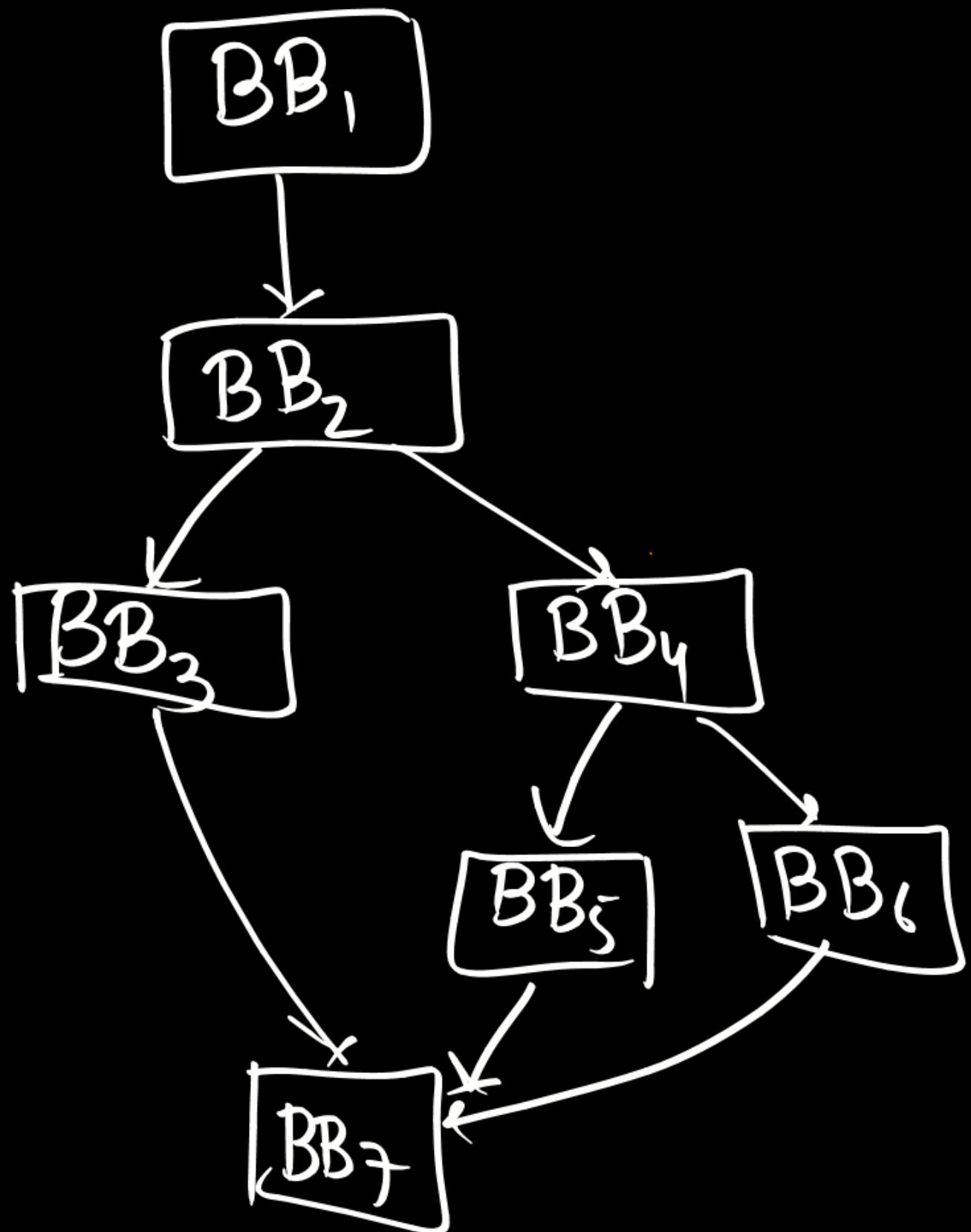
9 $x = x + b$

BB1
BB2
BB3
BB4

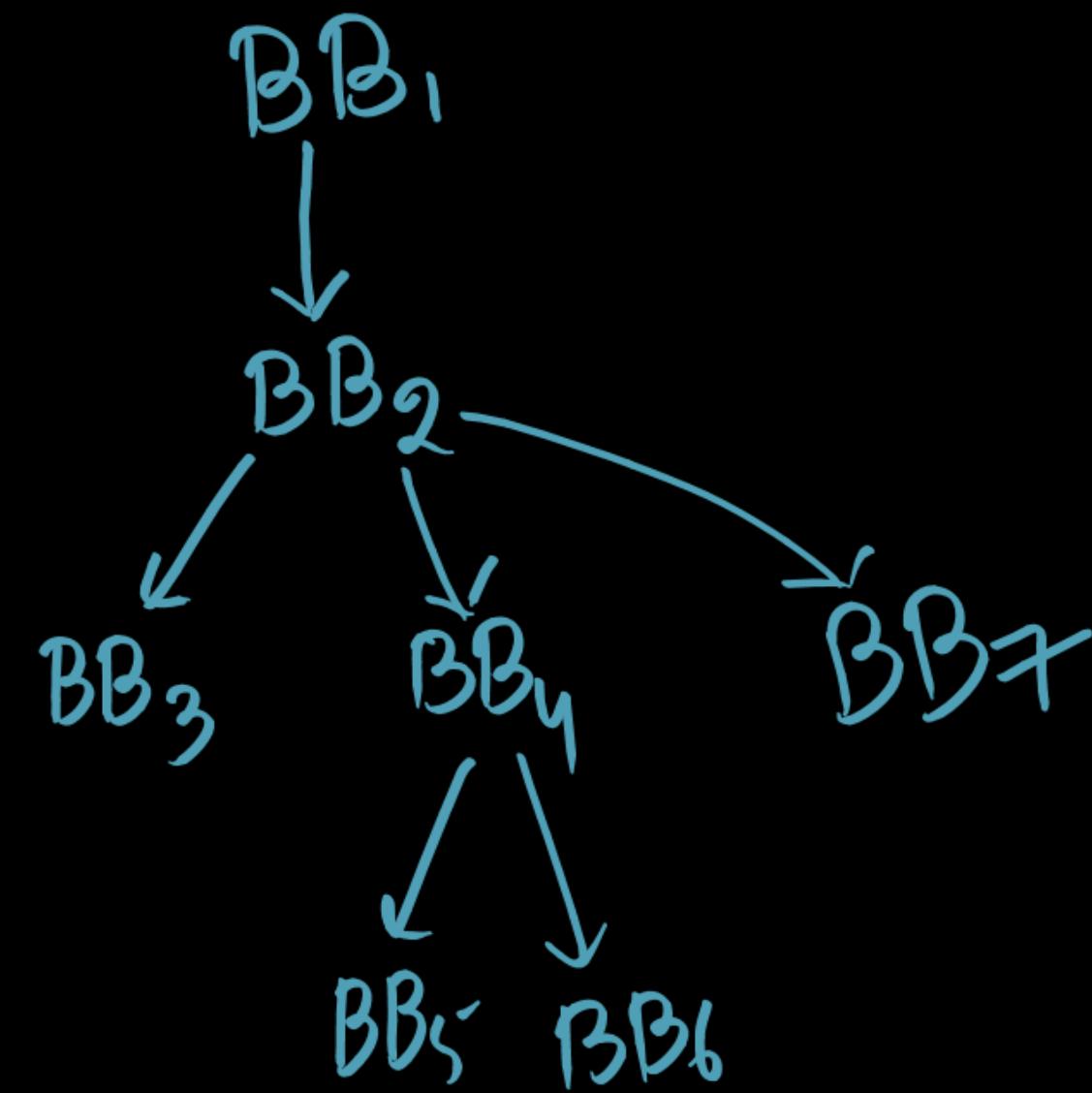
	GEN set	KILV set	OUT set	IN set
BB ₁	a, c	a, b, c	a, b, c	a, c
BB ₂	a, b, c	P, q, x	i) x, P, c, b ii) a, x, P, c, b iii) same	i) a, b, c ii) a, c, b iii) same
BB ₃	P, c, b	a, b	i) x, b ii) x, b, a, c iii) same	i) x, P, c, b ii) x, P, c, b iii) same
BB ₄	x, b	x	i) φ ii) a, b, c iii) same	i) x, b ii) a, c, x, b iii) same



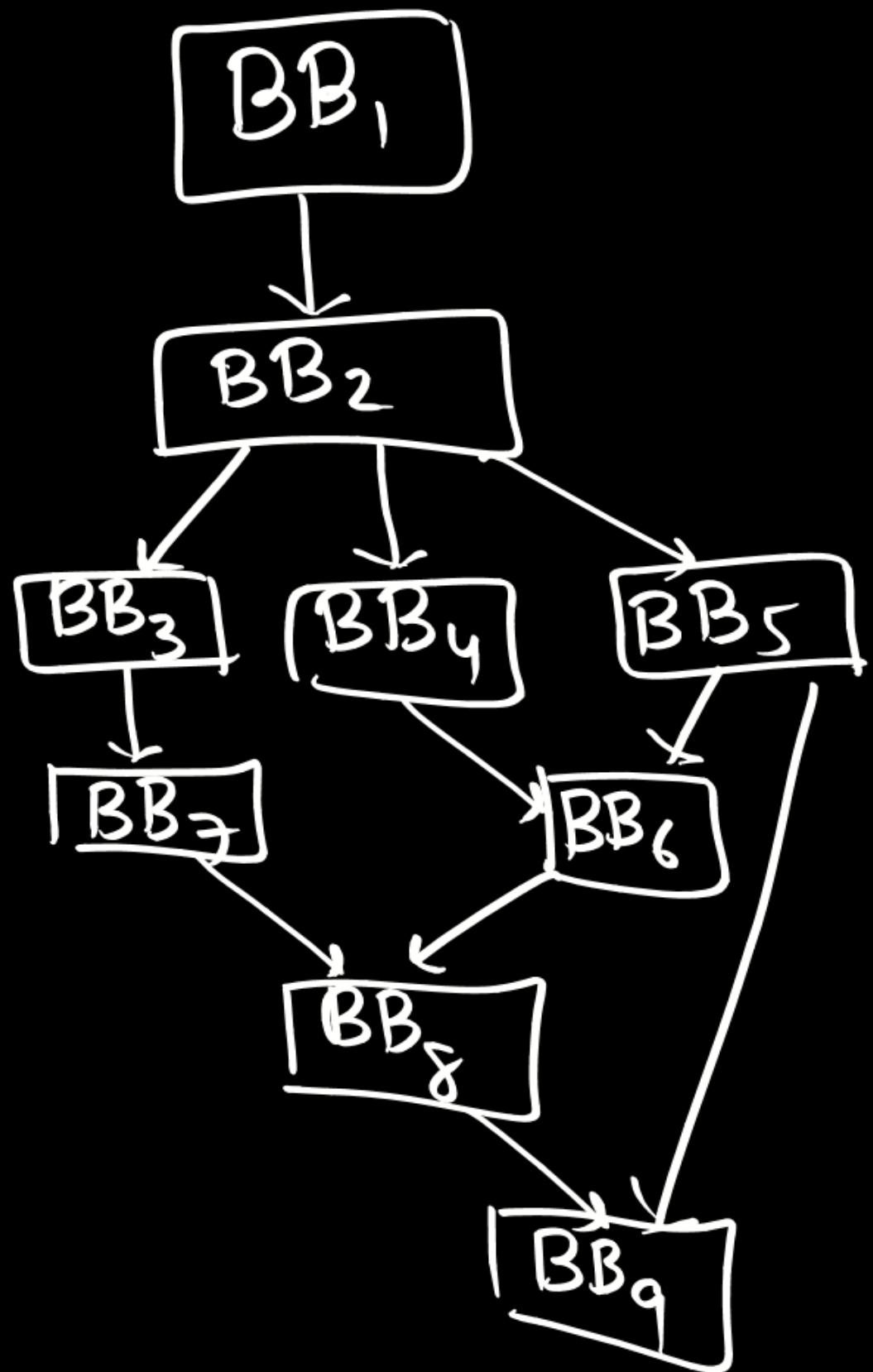
control Flow Graph:



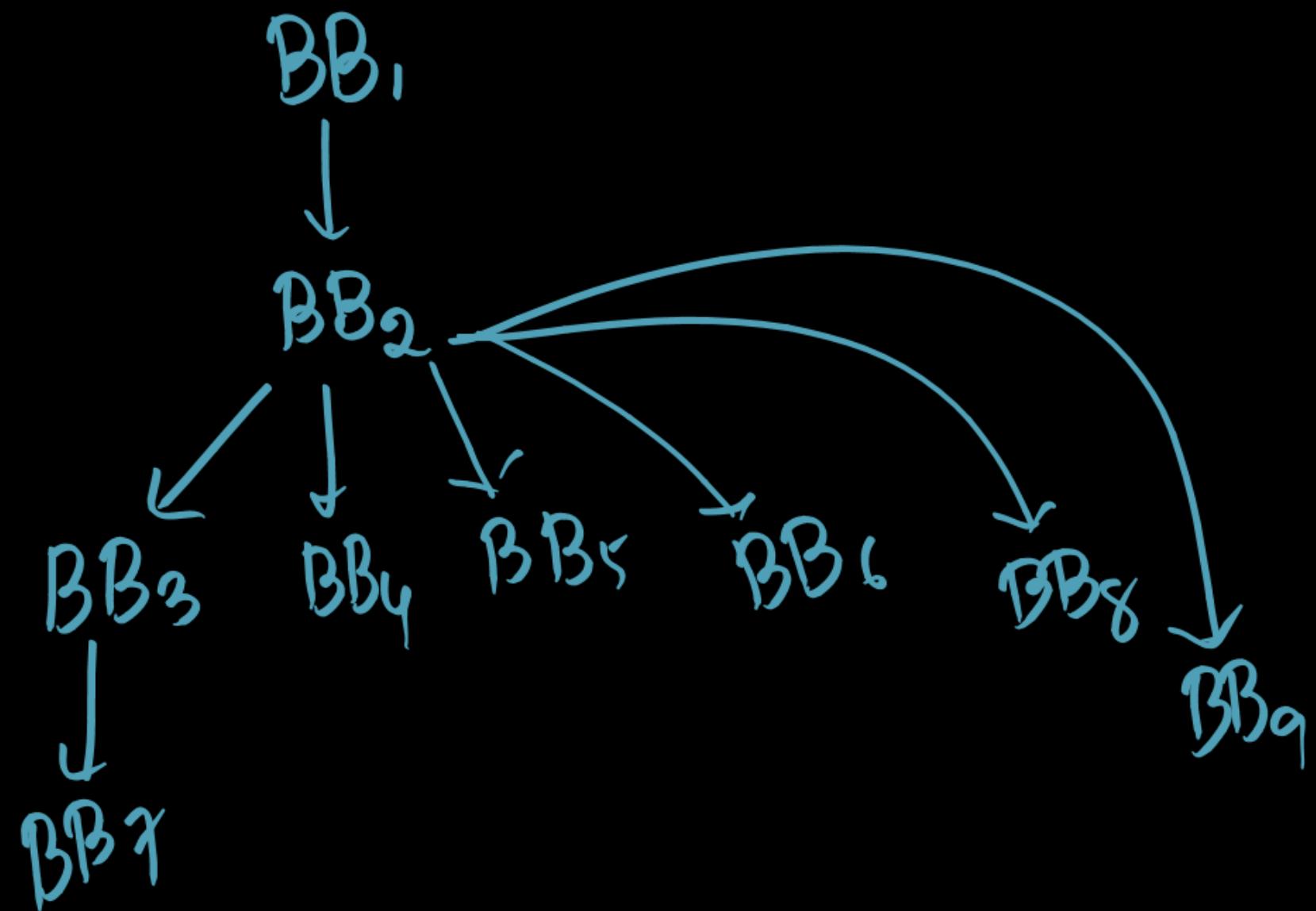
Dominator Tree



Control Flow Graph:



Dominator Tree

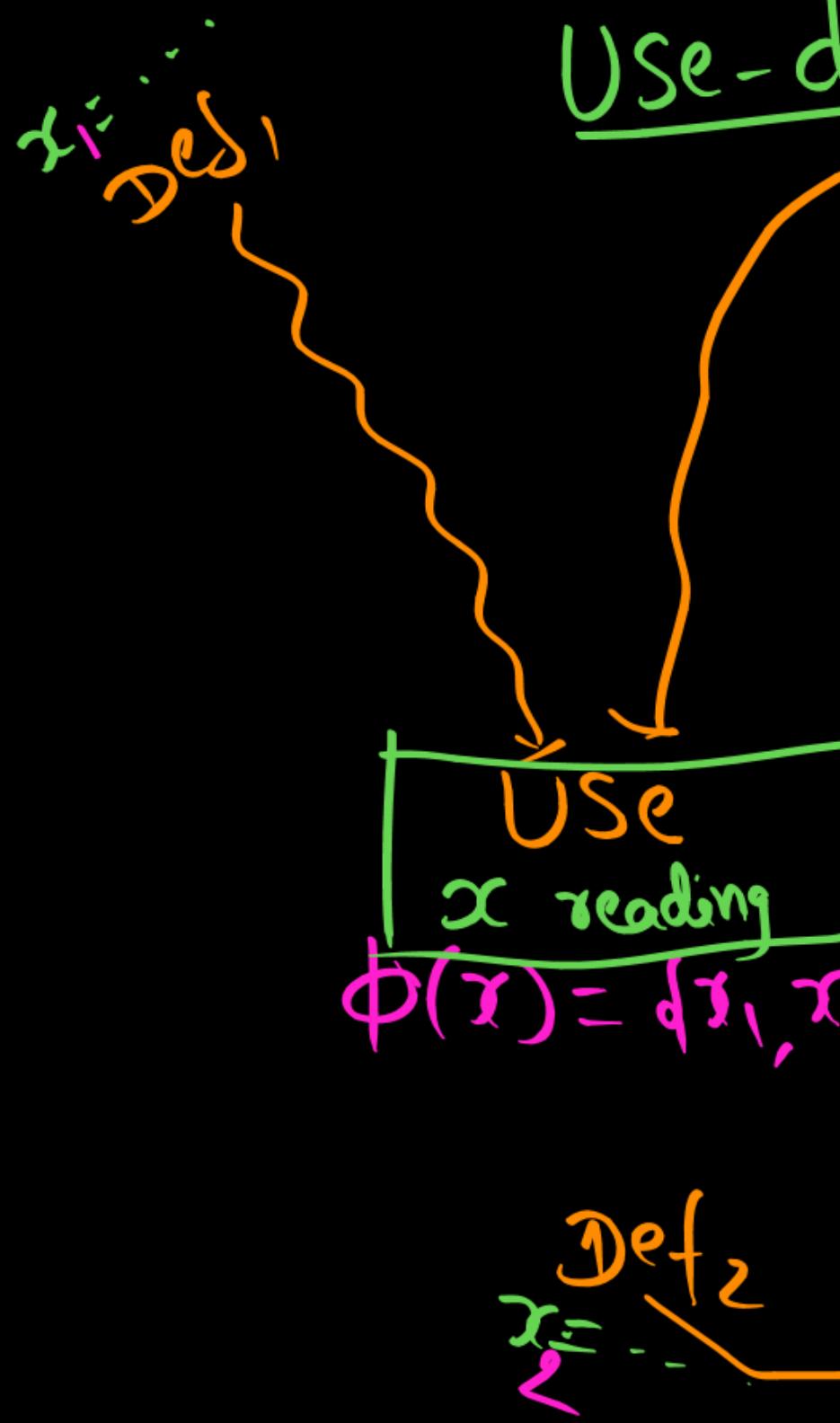


Reaching Definitions Analysis

I) **Use**- def_s chain

II) **Def**- use_s chain

Use-def chain



Def-use chain

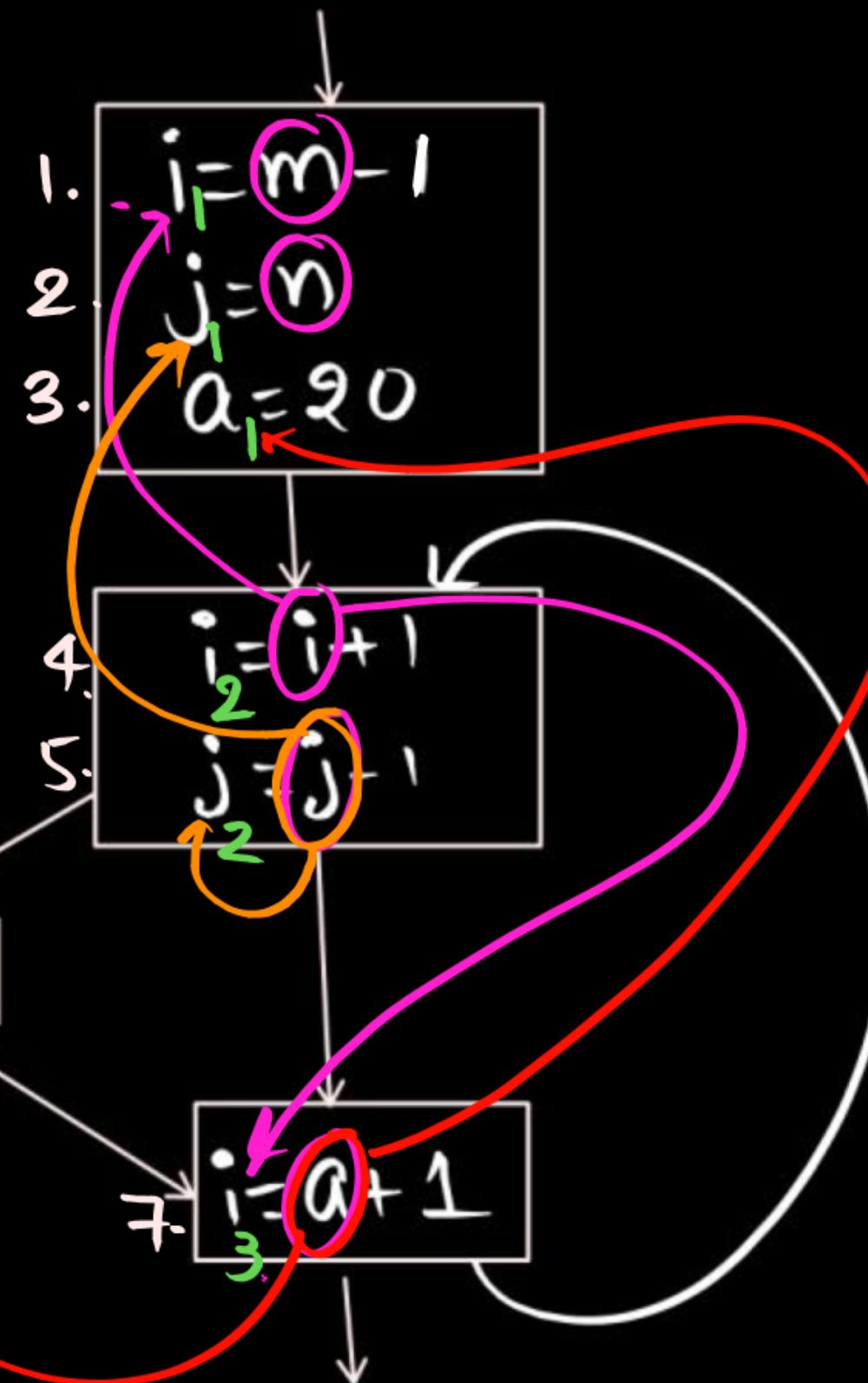


I) Use-Def chains

$\phi(j) = \{j_1, j_2\}$
at statement 5

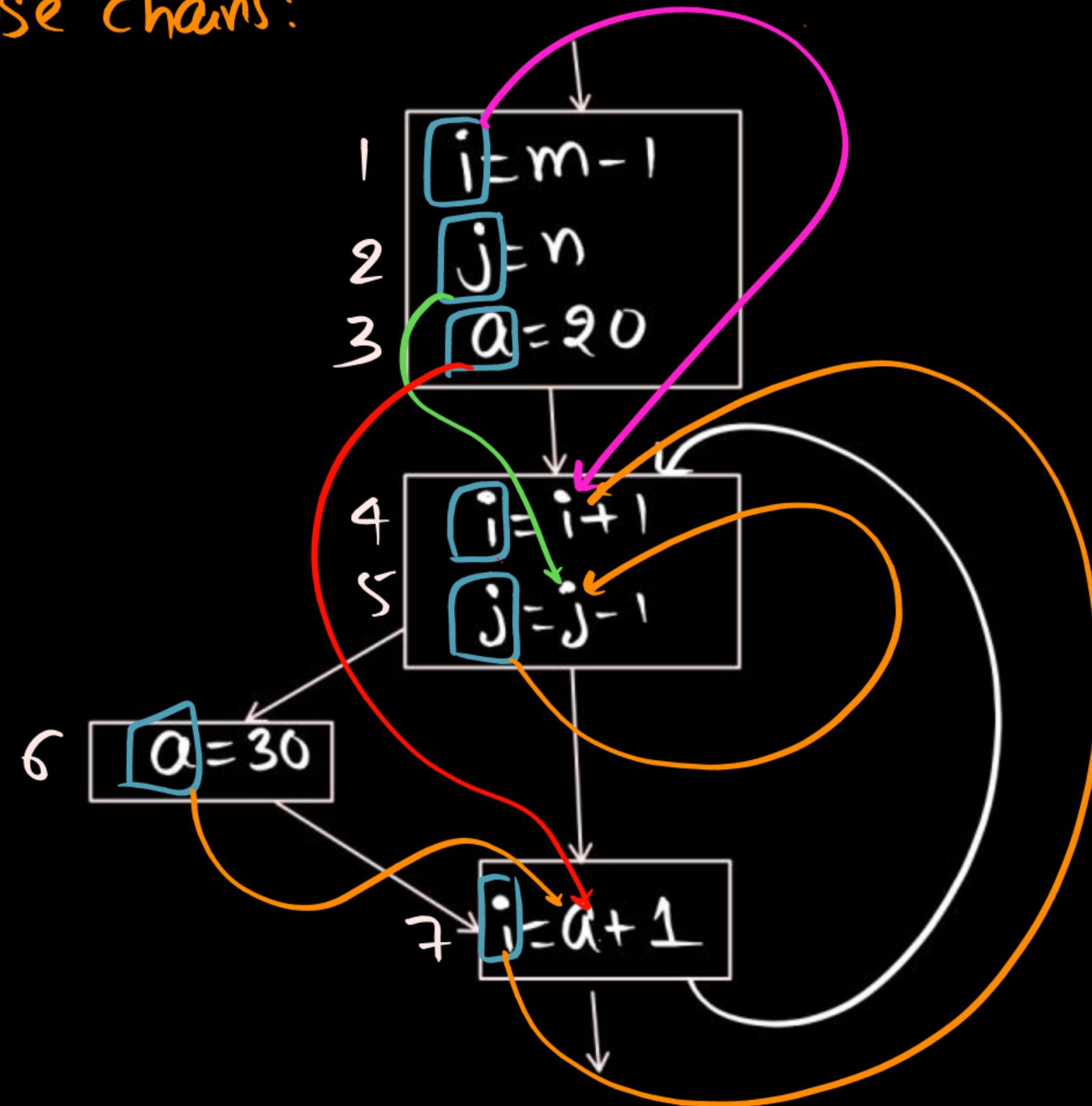
at statement 4
 $\phi(i) = \{i_1, i_3\}$

$\phi(a) = \{a_1, a_2\}$
at statement 7



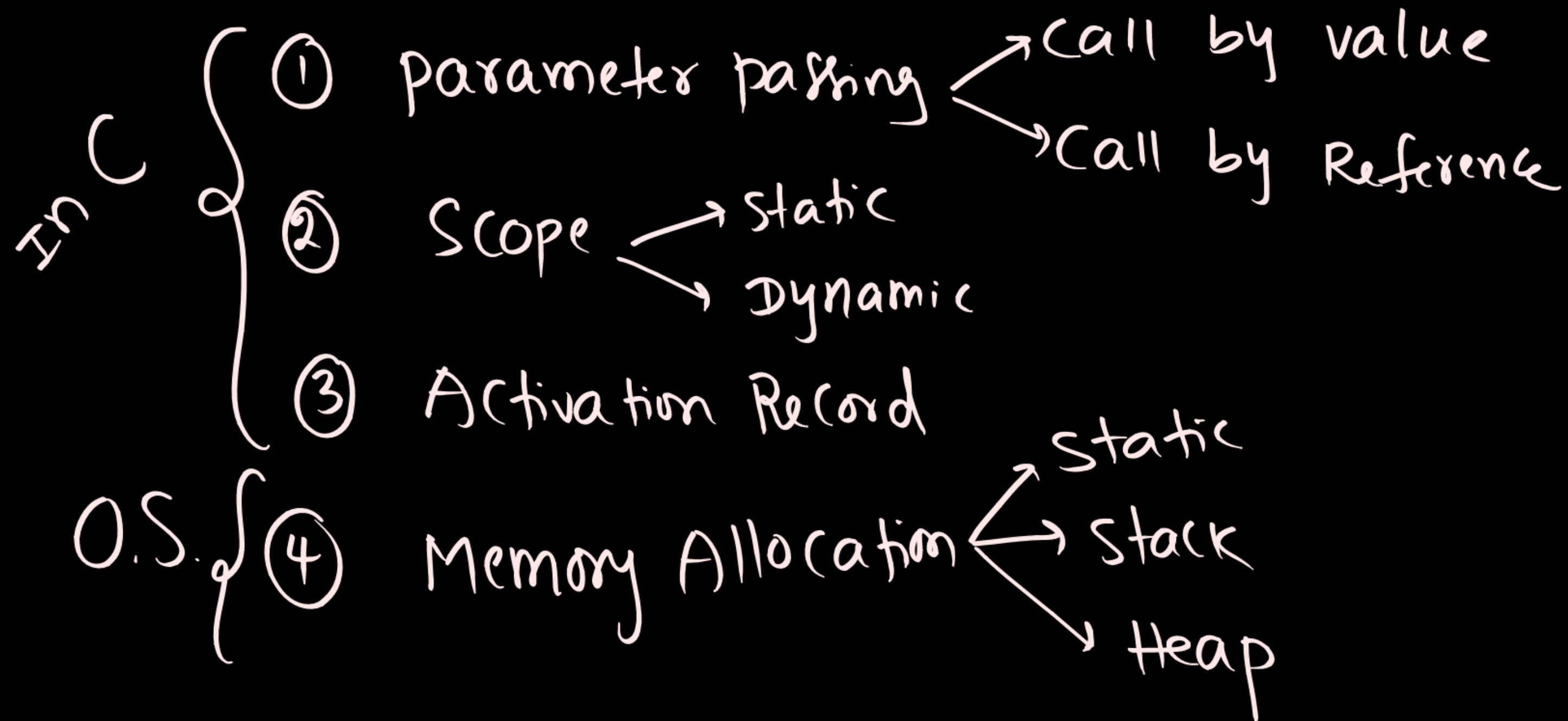
Statement	USE	Definition
1	m	-
2	n	-
4	i	i, j
5	j	-
7.	a	3, 6

II) Def-Use chains:



Statement	Def	Uses
1	i	4
2	j	5
3	a	7
4	i	4
5	j	5
6	a	7
7	i	4

Run Time Environments.



→ Compiler ✓

