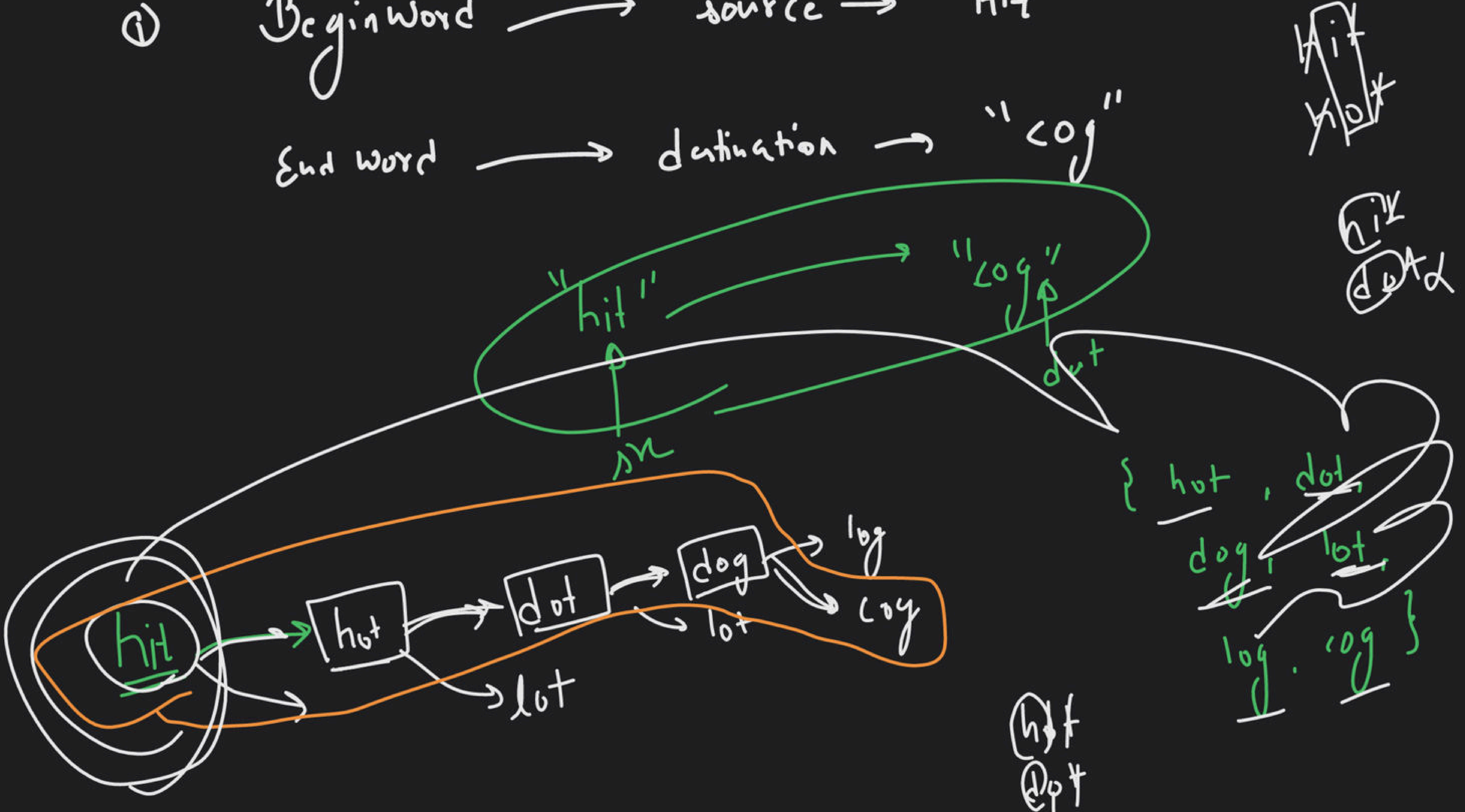
Solving Hard Problems using Backtracking - Level 3 & Doubt Clearing Session

Special class

Love Babbar • Oct 16, 2022

① Beginword ⟶ source ⟶ "hit"

End word ⟶ destination ⟶ "cog"



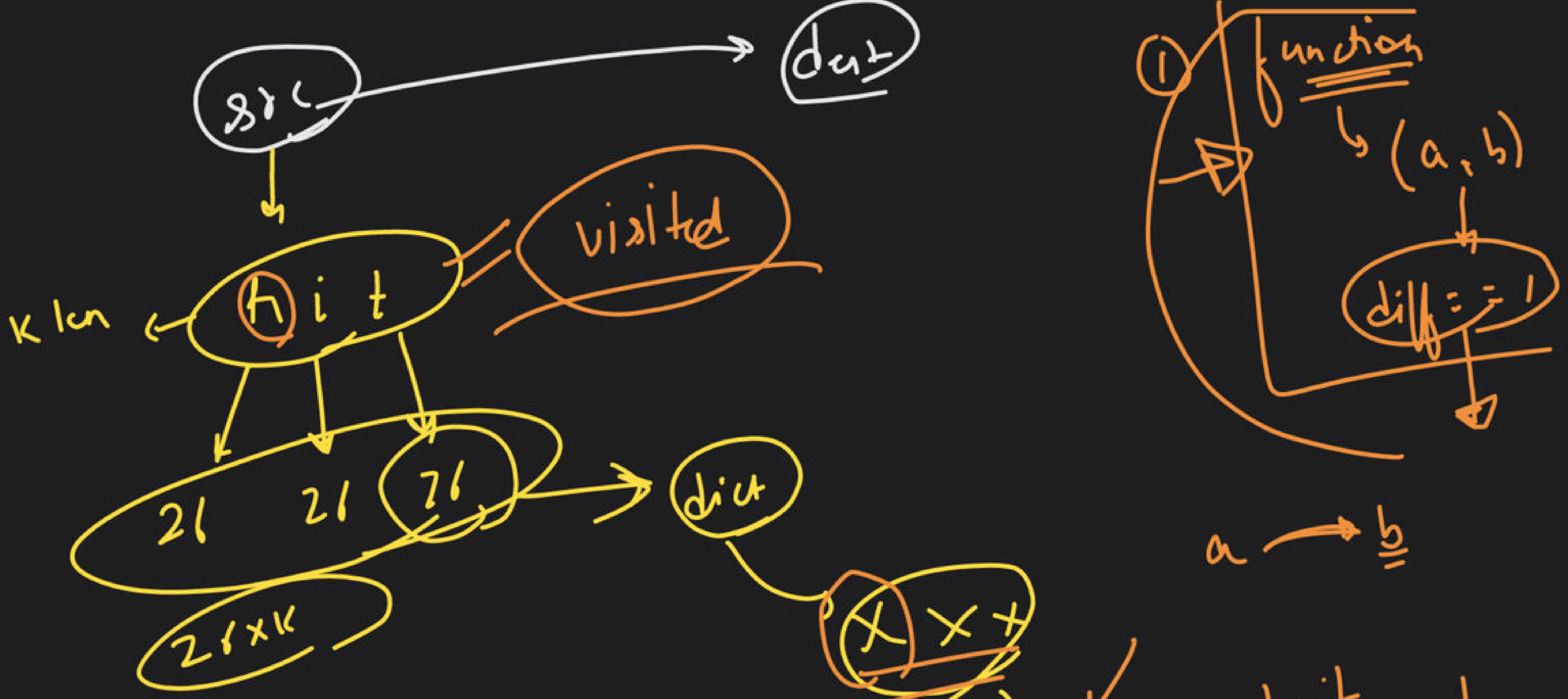
src → hit

dst → $\boxed{\text{cog}}$

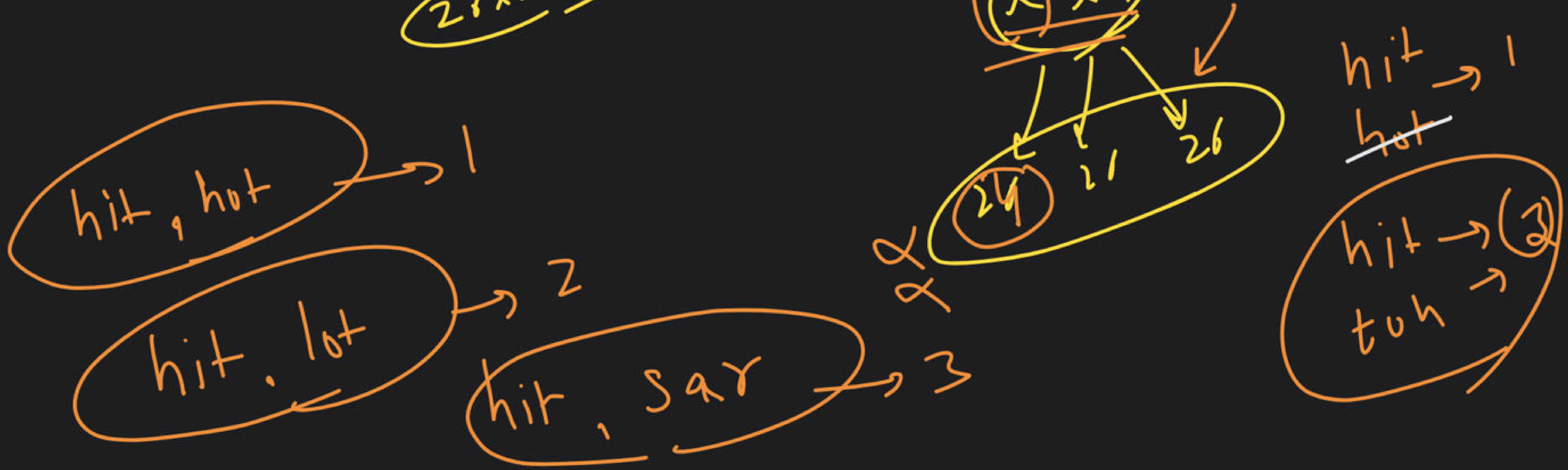
dict → { hot , dot
dog , lot , log }

[ ]

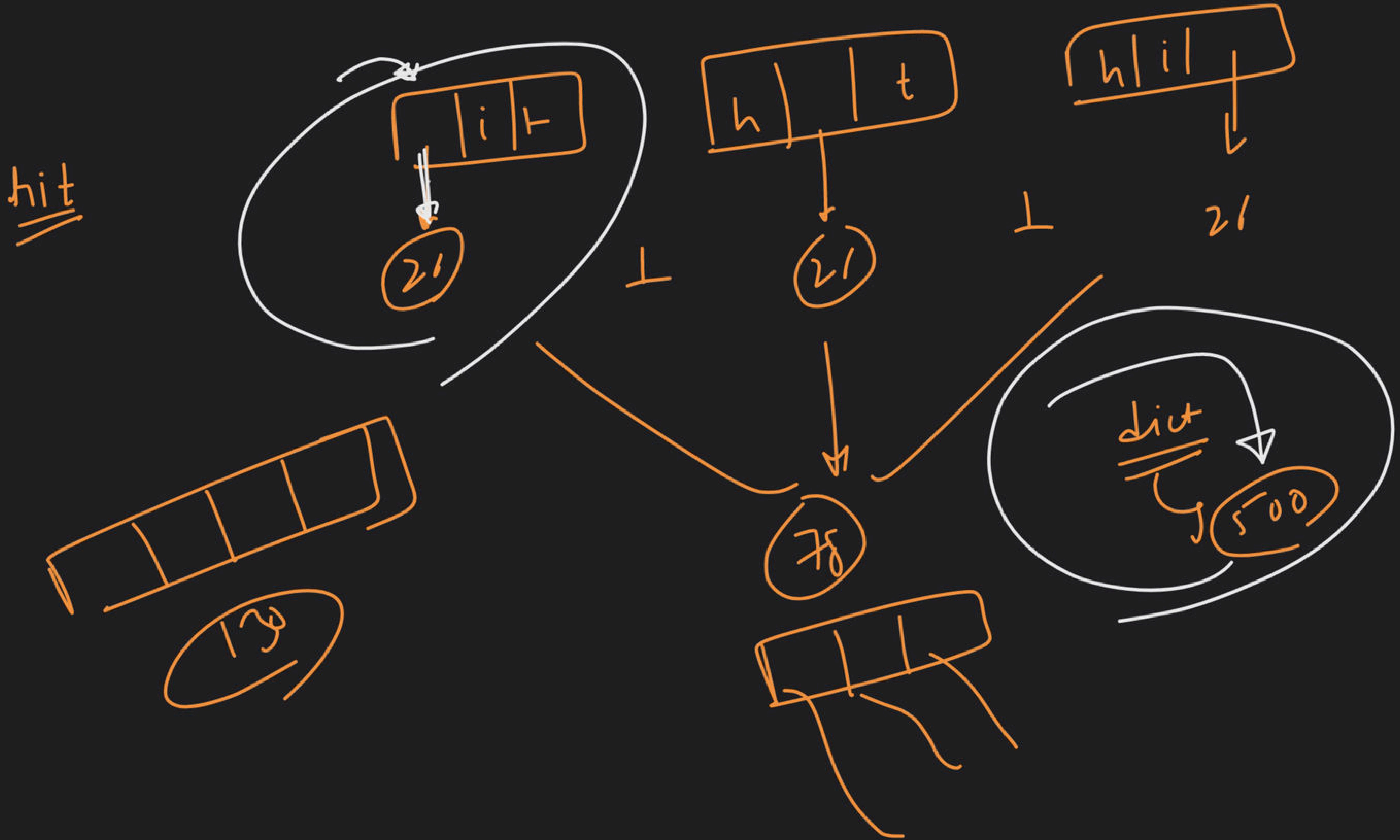
approach



src
dst
hit
k len
visited
26 26 26
26×k
dict
x x x
26 26 26
1
function
(a, b)
diff == 1
a → b



hit, hot → 1
hit, lot → 2
hit, sar → 3
hit → 1
~~hot~~
hit → 2
toh →



hit
i
r
21
h
t
h
i
21
⊥
⊥
21
78
130
dict
500

~~sit~~

hit

{ hot , lot ,
dot , dog ,
log , cog }

dot

```
for (auto w : dict)
{
        if (is Conn( _ , _ ))
        {
                if (!visited [w])
                {
                        ________
                        ________
                        ________
                }
```

Act
Re
Undo Action

```
solve ( ____________ )
{
    //B.C                         reached
    [          ] → rukna kab     destination
                                  ↓                        path
              h  path[path.size() - 1] == endWord
                                        ↘ ran path
                                        ↘ return;

                                              hit → hot
                                                  ↘ ~~dot~~

    for (auto w : dict)
    {
        if ( ________ && ________ )
        {
                  Action
                  RC
                  U.D
        }
    }
}
```

BackTracking + BFS → queue

map<string, bool> visited → T/F

or

set → erase

O(1)

→ build graph → BFS

→ path → src

→ solve ( )

→ return ans

set create

graph create

path → src

solve(

return ans

2 min

layer

1 diff

i/p → hit

{ hot, hit, sit }

{ hot, dot, lit, sit }

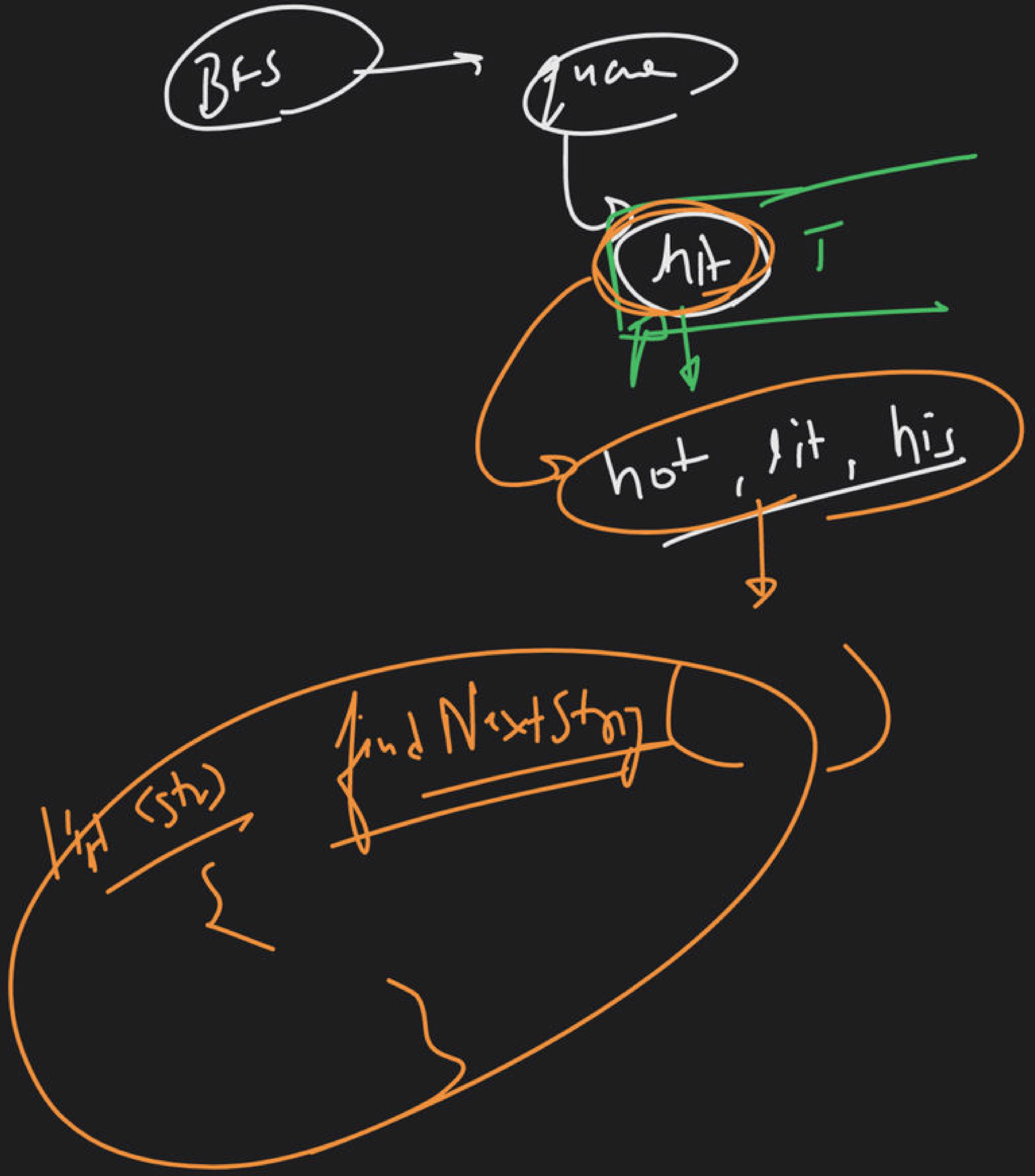
graph create

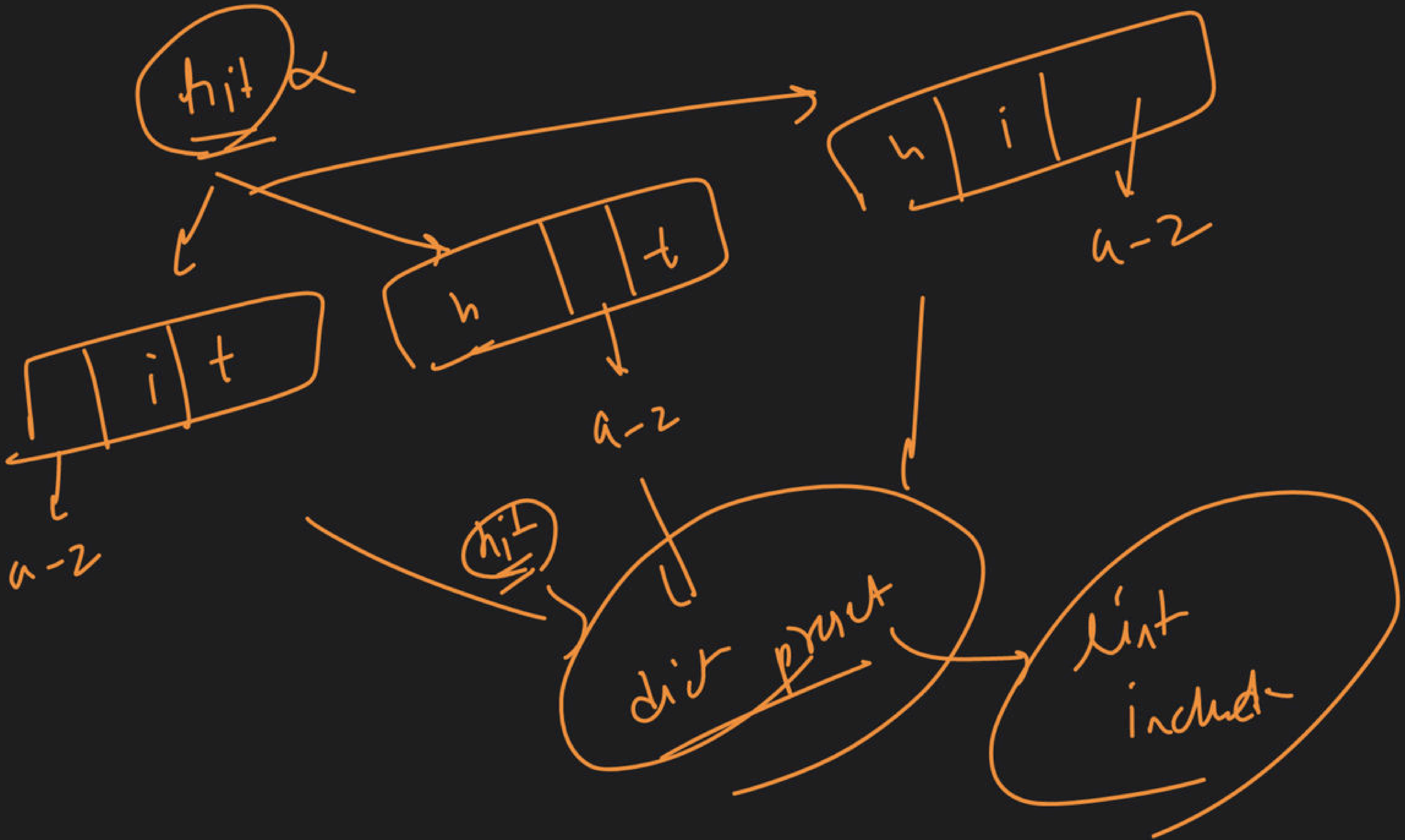
↳ adj list → fill

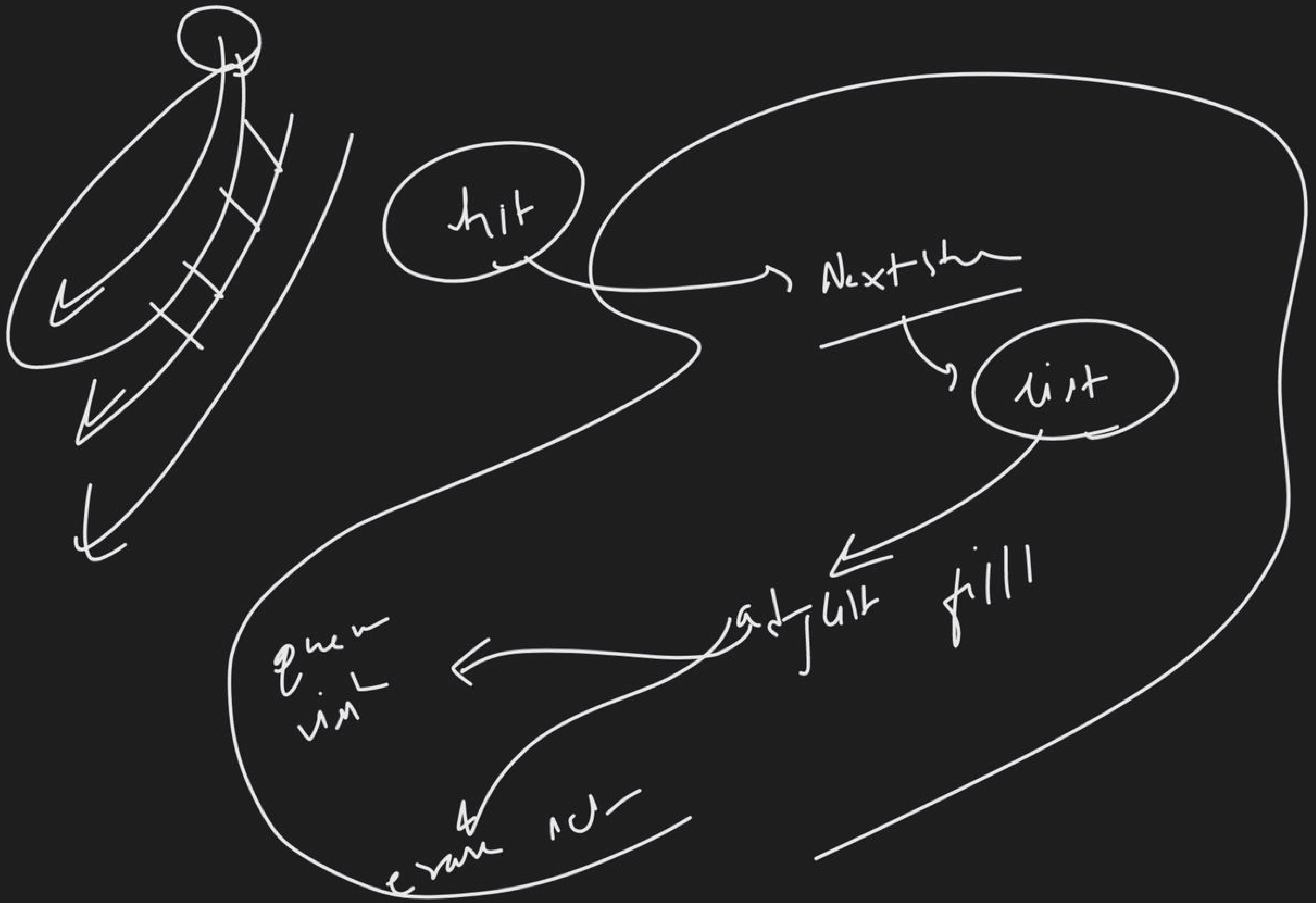
hot → hit,

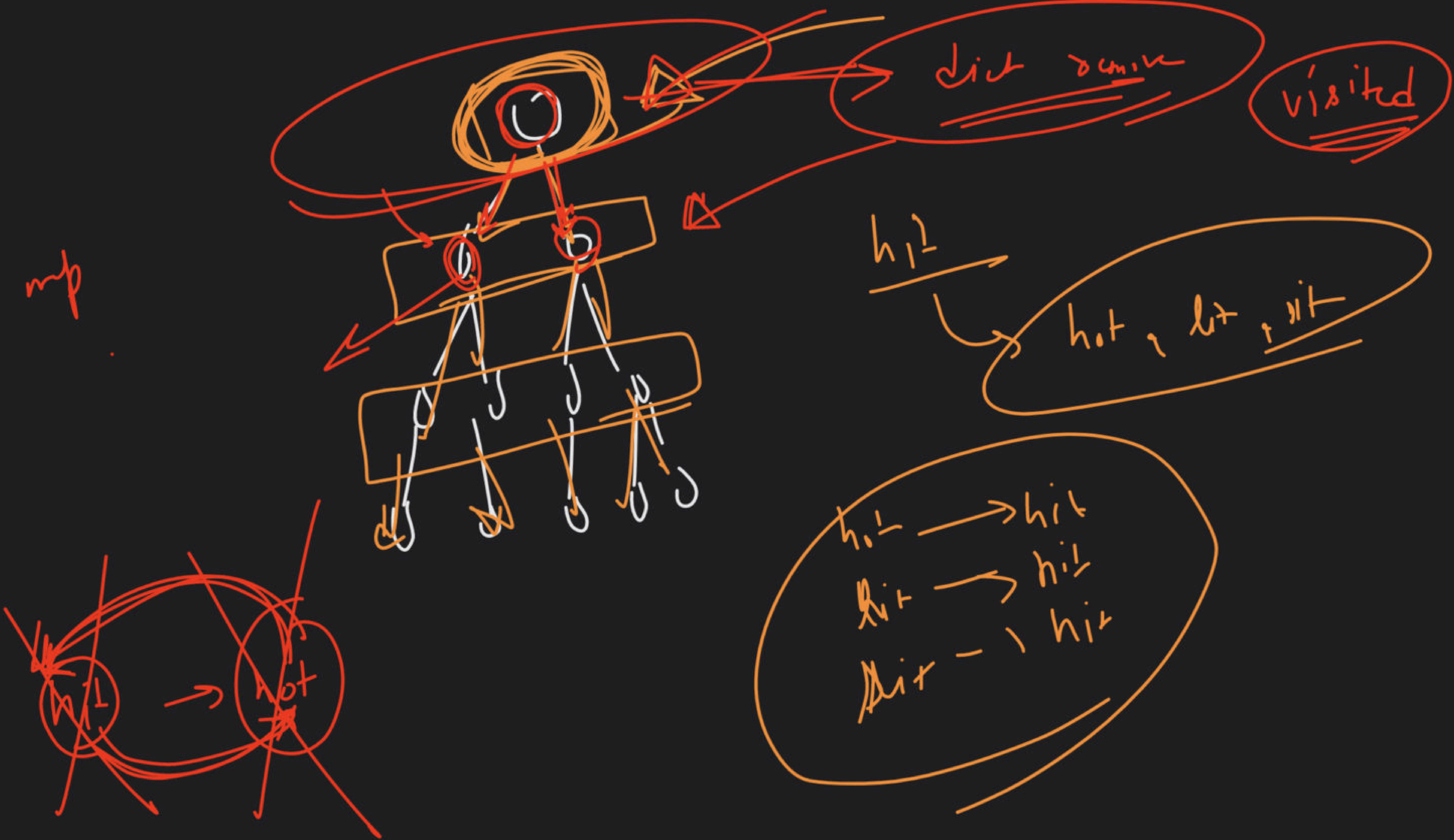
lit → hit,

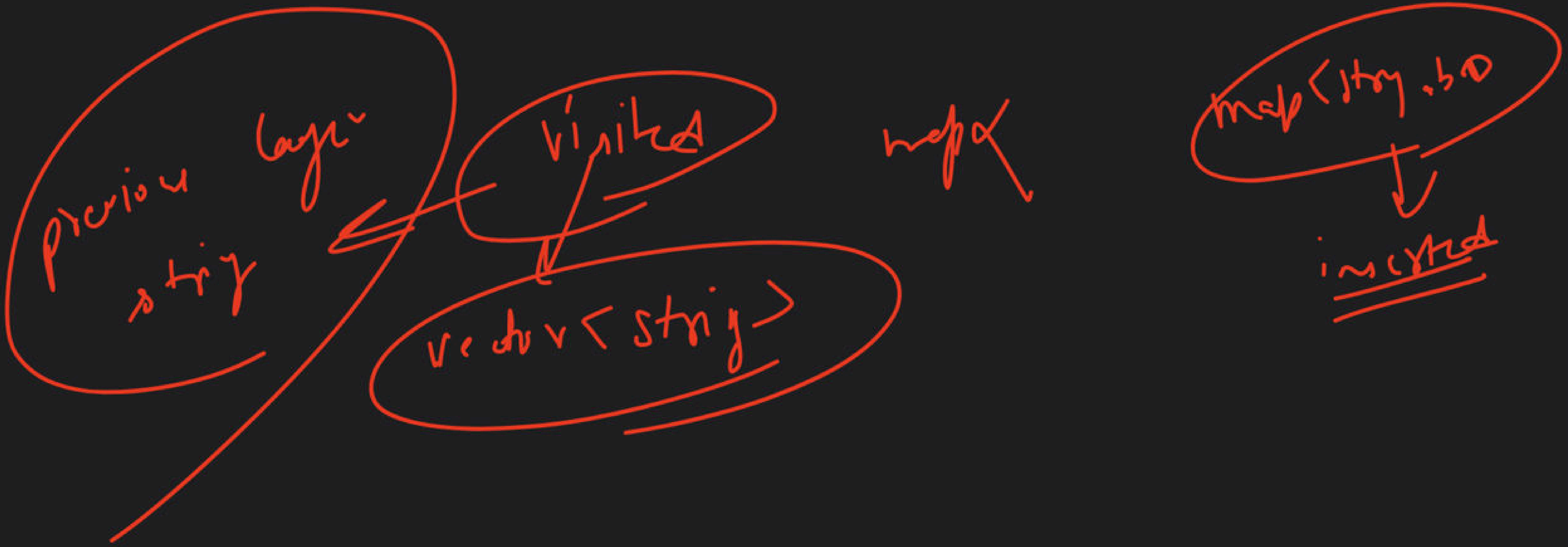
hin → hit,




hit
h
i
t
h
t
i
t
a-z
a-z
a-z
a-z
hit
dict pruct
list include


Init
Nextstate
List
adjust
fill
queue
visit
evaluate rule


dict remove
visited
mp
hit
hot, lit, dit
hot → hit
lit → hit
dit → hit
hit → hot


previous layer string
visited
vector<string>
~~map<~~
map<string, int>
inserted

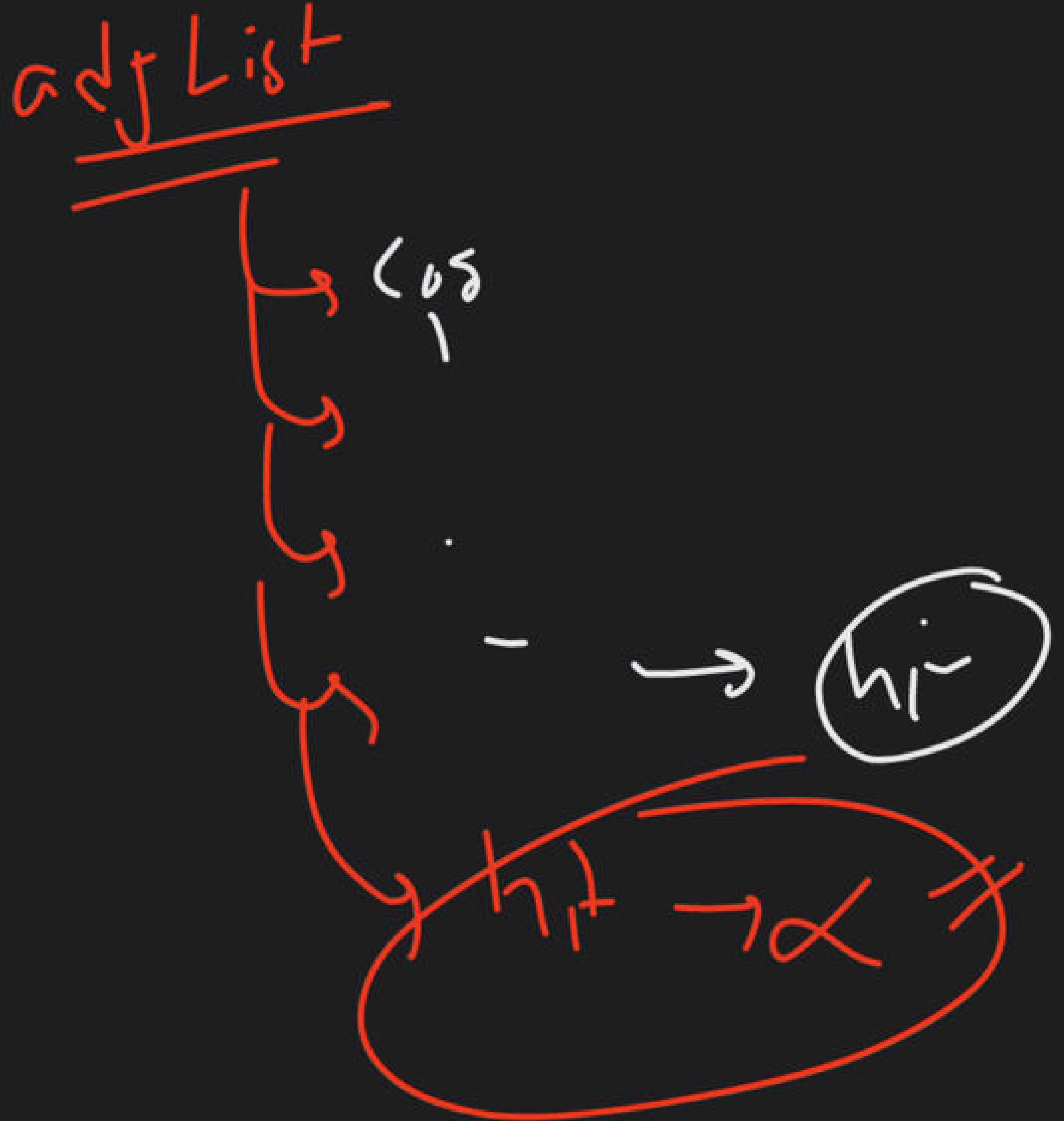
~~map < type , bool > m~~

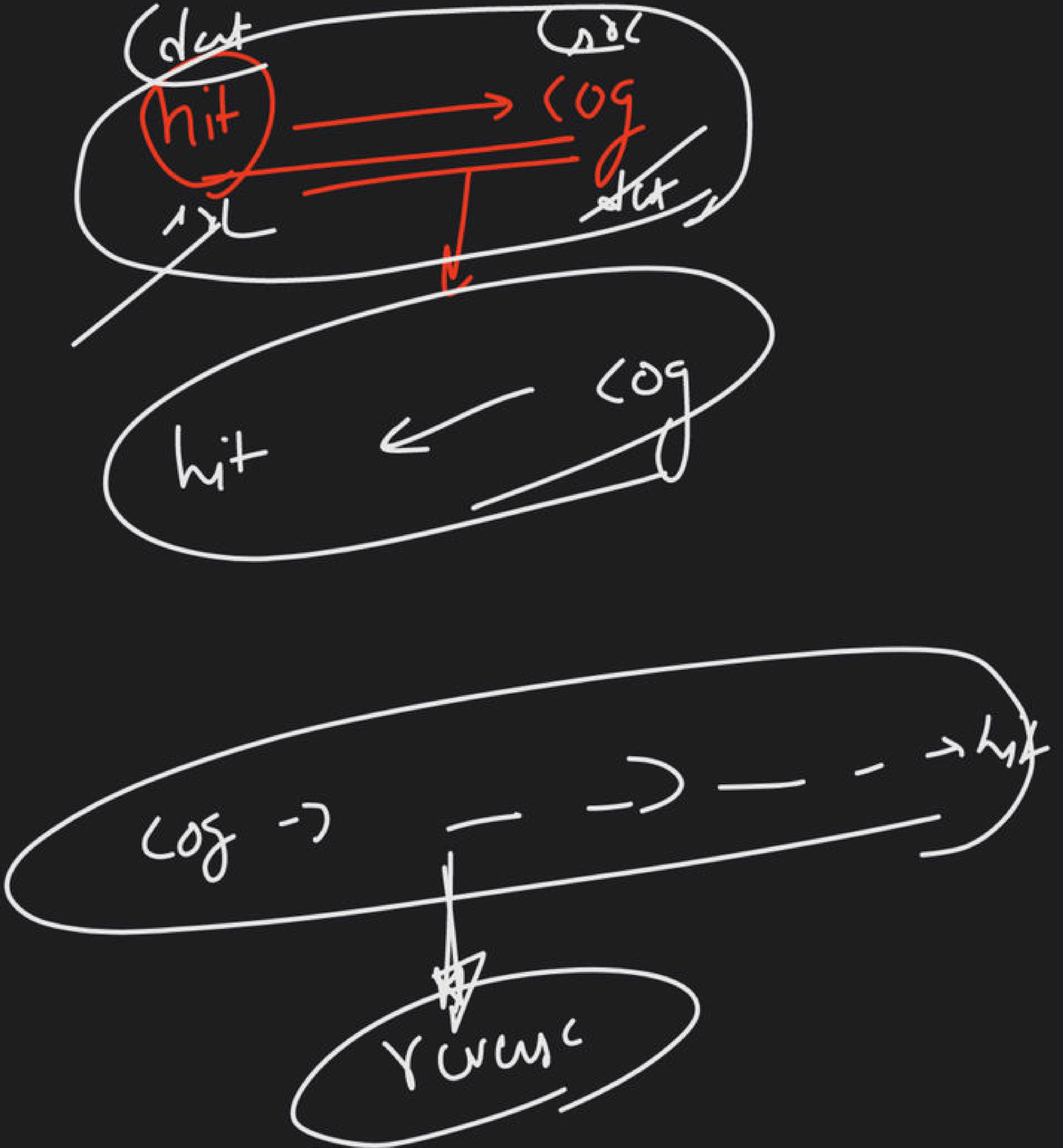
T F

set

~~unordered_set < string > s~~

~~s . erase ( )~~


adjList
cog
hit
hit → ✗


dest
src
hit
cog
src
dest
cog
hit
cog →
→ hit
reverse

→ Code → 5 Ex → Dry Run

↓

T·C → O(Nk²)

↓ ?

How?

→ 50 Q → 75+ Q

H/W

H/W

(1)

→ 17 → 20+

→ 20. Wed → 9 → 11 pm

↓

D·S → Last Class

→ DP

$$\begin{tikzcd}\boxed{\text{Code}} \arrow[r] & \boxed{75\%}\end{tikzcd}$$

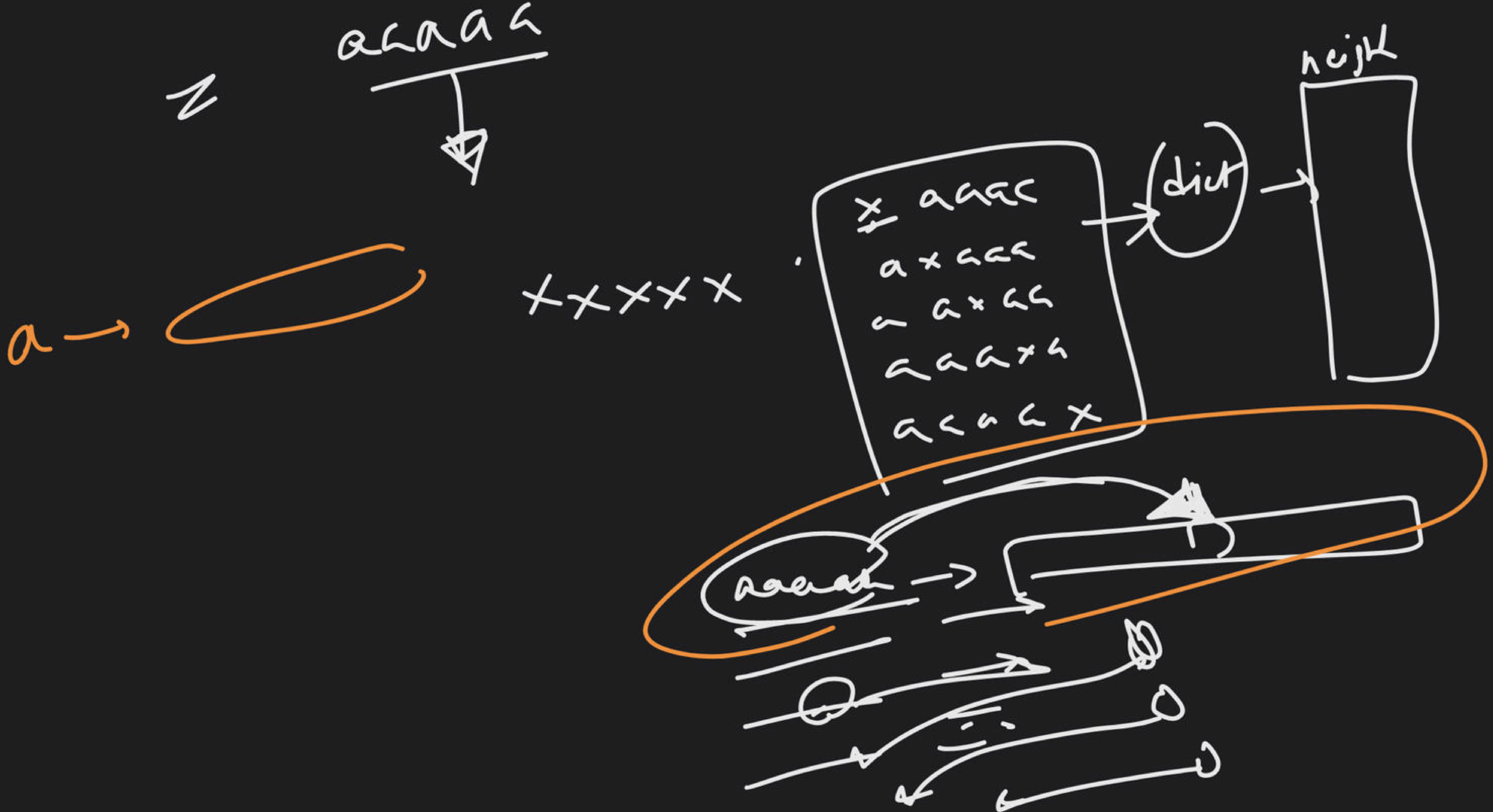
Graph

DSA

WebDev

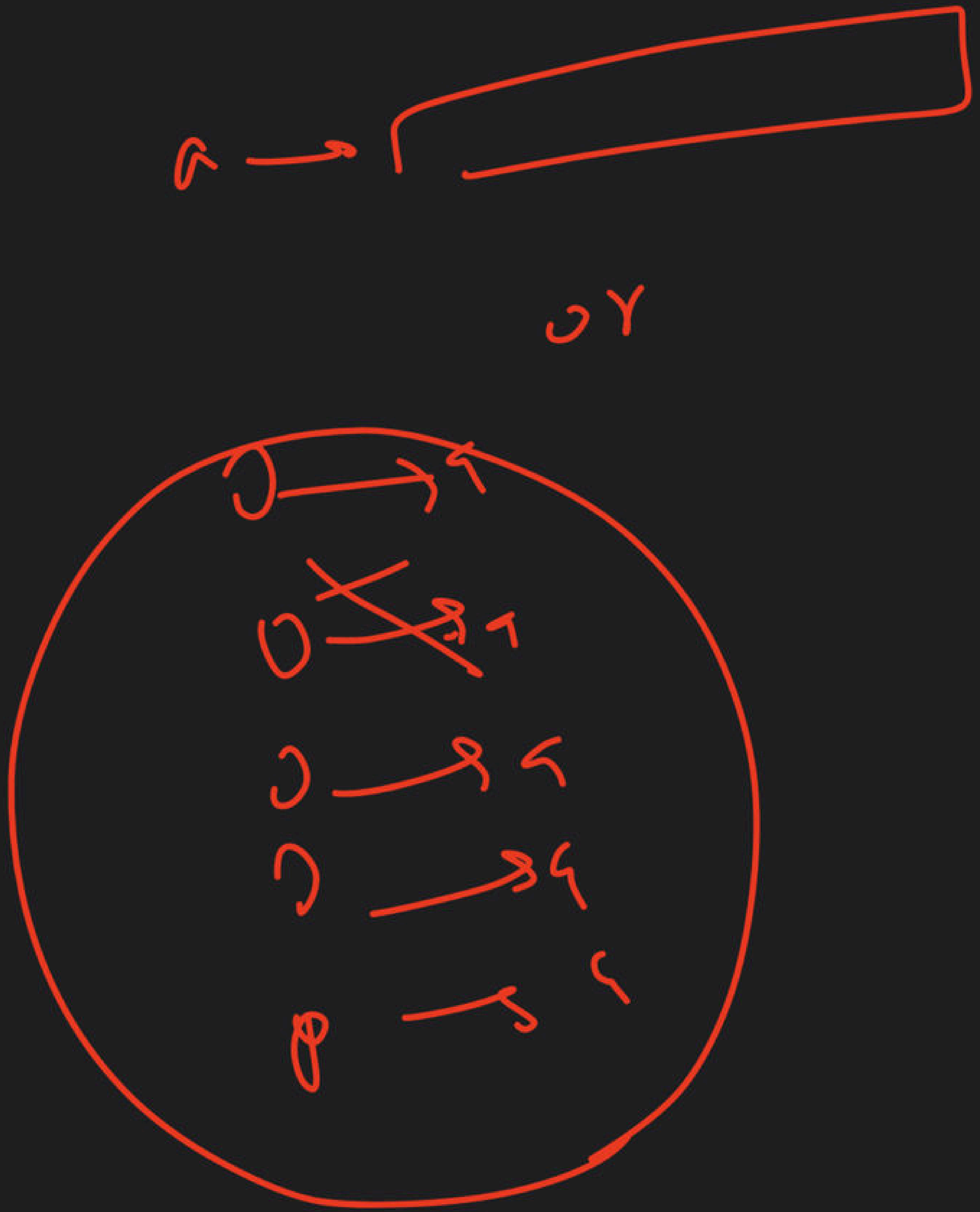
$$\begin{tikzcd}\text{IPB} \arrow[d] \\ \text{R.V} \arrow[r] & \text{1 day}\end{tikzcd}$$

IAS



aaaaa
z
a →
xxxxx
x aaac
a x aaa
a a x aa
a a a x a
a a a a x
dict
neigh
aaaah

a


a
or

dst

1 Xtra D.S $\longrightarrow$ Wed $\rightarrow$ $\boxed{\text{9 - 11pm}}$