Agenda → We will discuss a lot of matrix based recursion

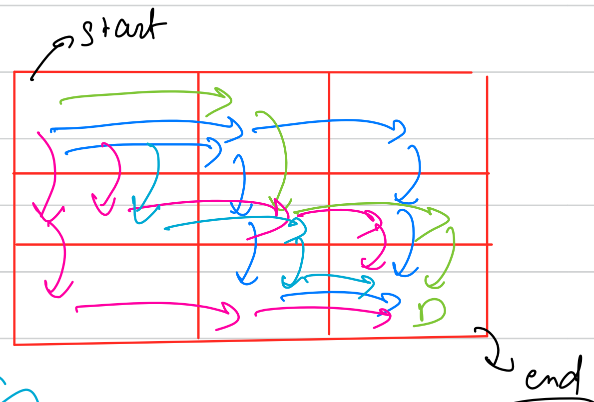
→ Advanced recursion

Qn You are given a 2D matrix of dimension $(n \times m)$

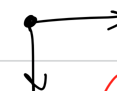
You will start from top left cell & you want to go to bottom right. At each point you can move either right or down. Print & count all possible paths

R R D D
R D D R
R D R D
D D R R
D R R D
D R D R

6



3x3



Input → (n, m)

1) Base Case
2) Recursive Intuition
3) Self work

What if you can also move diagonally

start $(0,0) \longrightarrow$ end $(n-1, m-1)$

Top left $\longrightarrow$ Bottom right

$\boxed{(i, j)} \rightarrow$ right

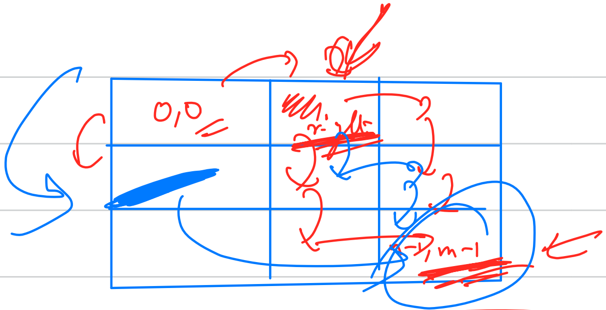
down

if $i >= n$ and $j >= m$

// come back

$(i == n-1$ and $j == m-1)$

current cell is the dest cell



D DRR
D RRD
D RDR

1/2

correctly

R ROD
R DDR
R DRD

1/2

$f(i, j, osf) \longrightarrow f(i+1, j, osf + \text{"d"})$

$f(i, j+1, osf + \text{"R"})$

self check

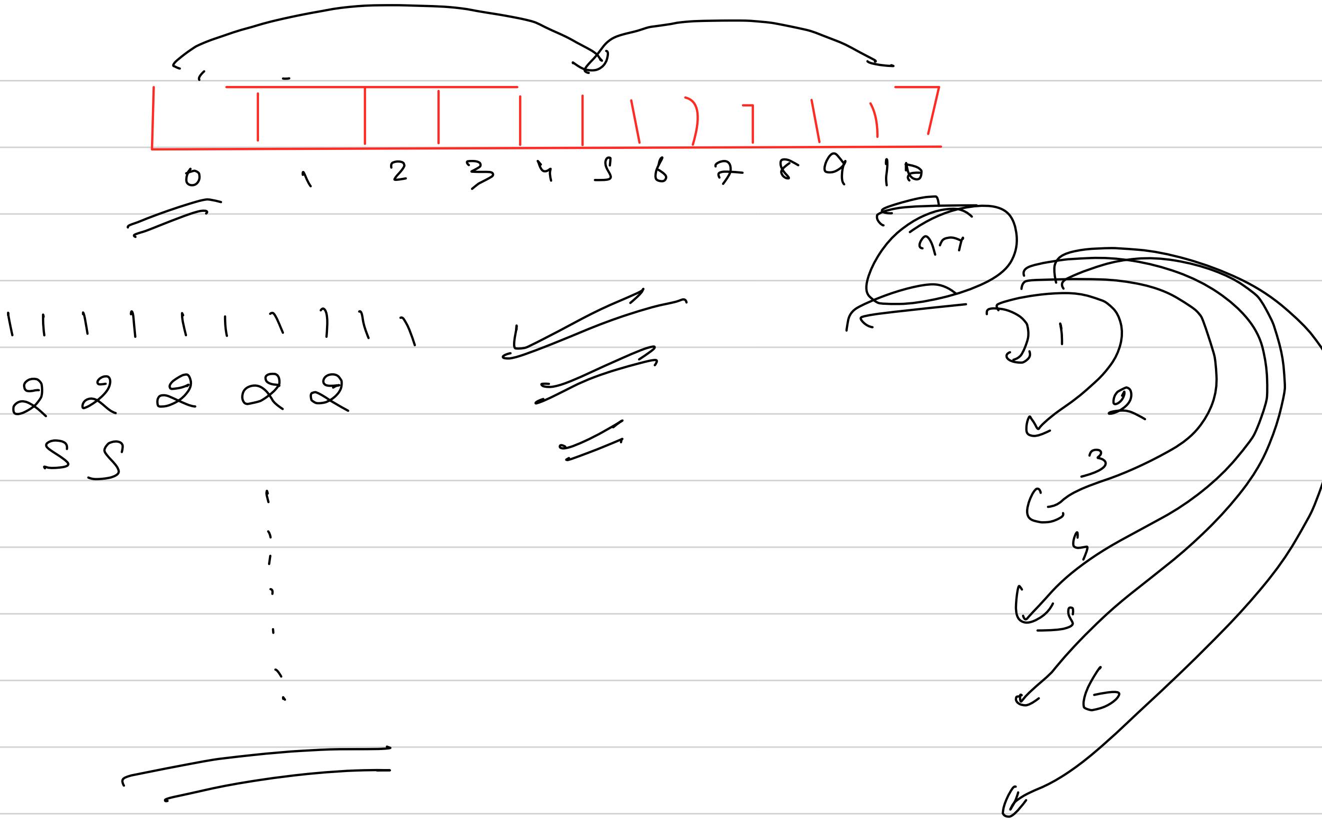
give all paths from $i,j \to n-1, m-1$

Q→ There are n cells arranged in linear fashion.

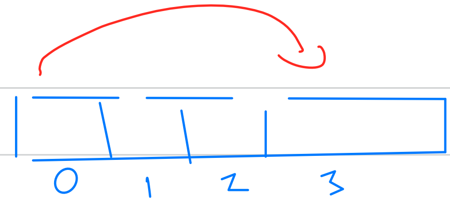
You are standing at the 0^{th} cell & want to reach $(n-1)^{th}$ cell.

At each cell you can have 6 possible jumps to make

In How many ways you can reach to $(n-1)^{th}$ cell.
Print the ways



find all paths



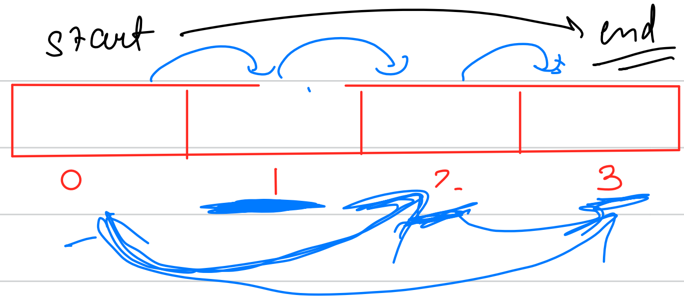
1 1 1

2 1

1 2

3

4 x 5 x 6 +



(4 length)

Start → S + 1, S + 2, S + 3, S + 4, S + 5, S + 6

jump on these 6 cells

1, 1, 1

1, 2

2, 1

3

Base case

$(i == n-1) \longrightarrow$ valid path

$(i >= n) \longrightarrow$ invalid paths

Recursive intuition $\Rightarrow$ Create all the paths from

$\{ \underline{s+i} \longrightarrow \underline{end} \quad \forall \; i \in \underbrace{[1, 2, 3, 4, 5, 6]}\}$

Self work $\rightarrow$ If you already got the paths from

$s+i \longrightarrow end$, we will just print $\underline{i}$.

Recursion

Q → You will be given a value n, print the first n+1 whole numbers in lexicographical order

n = 1000

→ dictionary order

1 → 2 ~~X~~

0 → A
1 → B
2 → C
3 d
4 e
5 f
6 g
7 h
8 i
9 j

0 → 1 → 10 → 100 → 1000

→ 101 → 102 → 103 → 104

- - - - - . 109 → 11 → 110

110 → BBA

$n = 13$

$$\left.\begin{array}{l} 0 \\ 1 \\ 10 \\ 11 \\ 12 \\ 13 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array}\right\}$$

correctly

i is no variable part

f(n, i)

Starting value of i → 0

print i → n

(lexicographical)

1) Base Case if (i > n) stop here!!

2) Recursive task

3) Self → print i

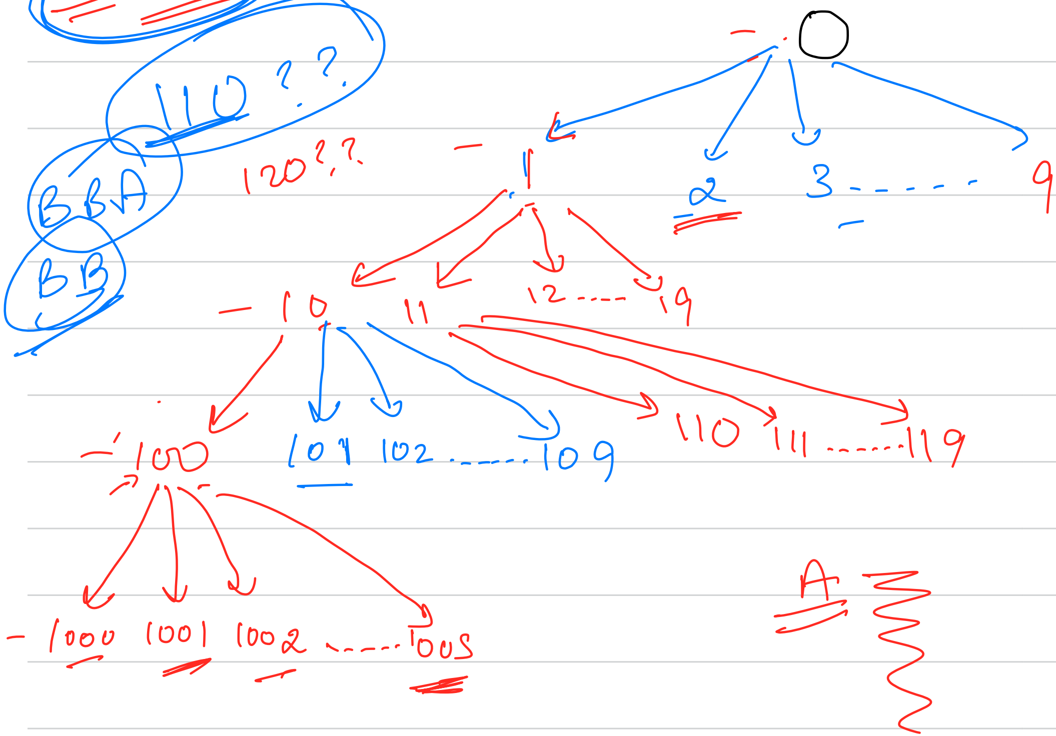
n = 1005

110 ??

120 ??

BBA

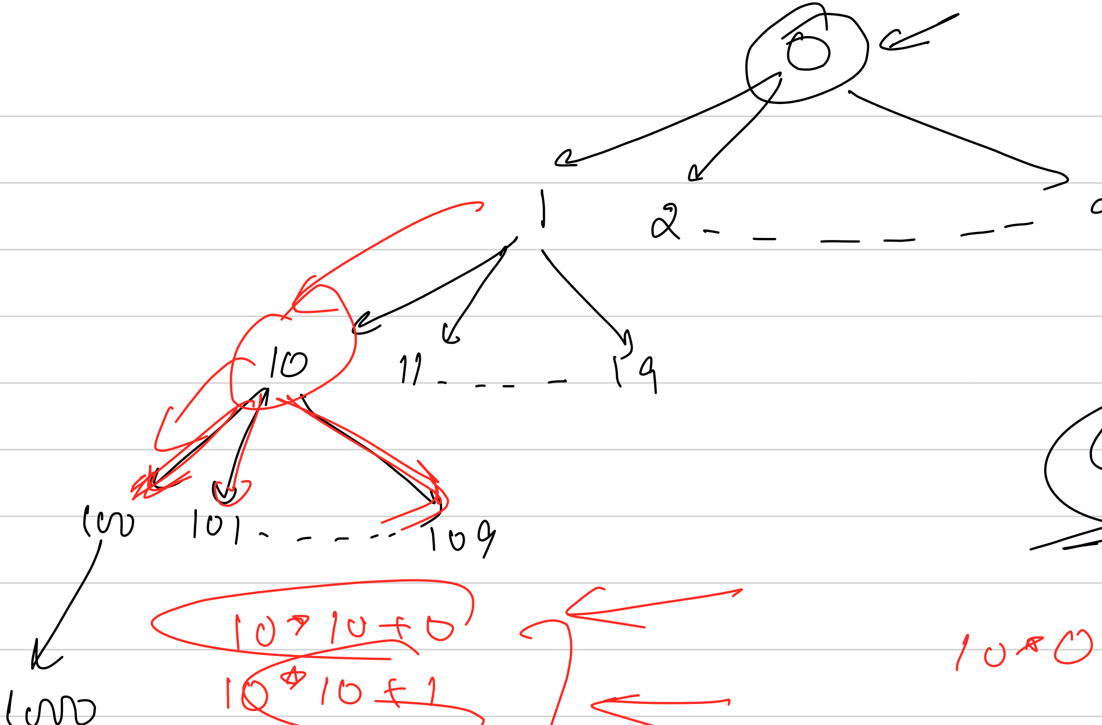
BB



n = 2

n = 13

0
1
10
11
12
13
2
3
4
5
6
7
8
9



$10*10+0$

$10*10+1$

$10*10+2$

$10*10+3$

$\vdots$

$10*10+9$

$0 \to n$

$10*0+1$

$10*1+0$

$10\times10+0$

$10\times10+$

$10*i+j$

Q:- Given a string, with all unique characters, print all possible permutations of string.

Ex ABC

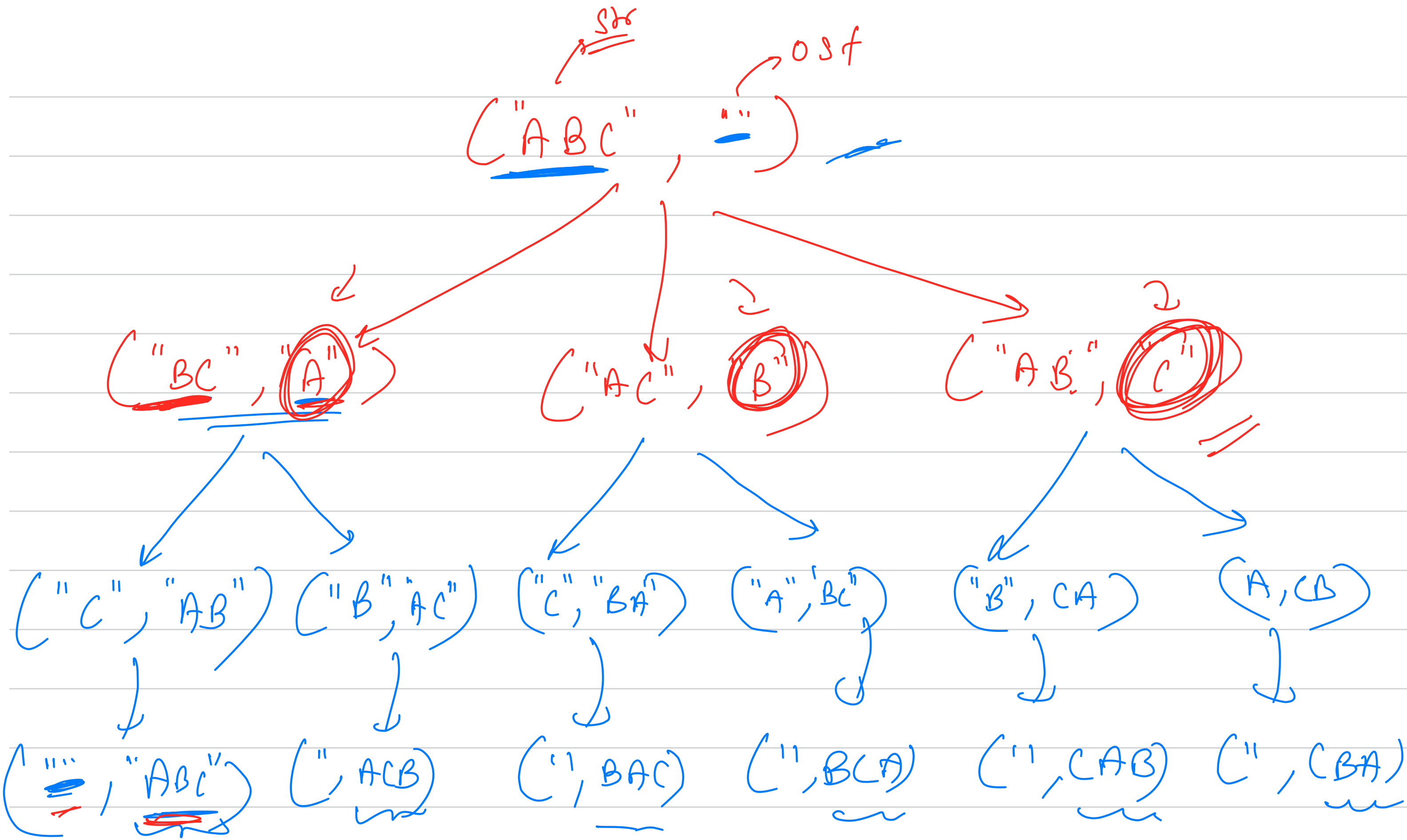
ABC
ACB
CAB
CBA
BCA
BAC

print

Recursion

2 mins

Backtracking



str
osf
("ABC", "")
("BC", "A")
("AC", "B")
("AB", "C")
("C", "AB")
("B", "AC")
("C", "BA")
("A", "BC")
("B", CA)
(A, CB)
("", "ABC")
("", ACB)
('', BAC)
('', BCA)
('', CAB)
('', CBA)

(ABC , "")

(BC , A)   (AC , B)   (AB , C)

A B C

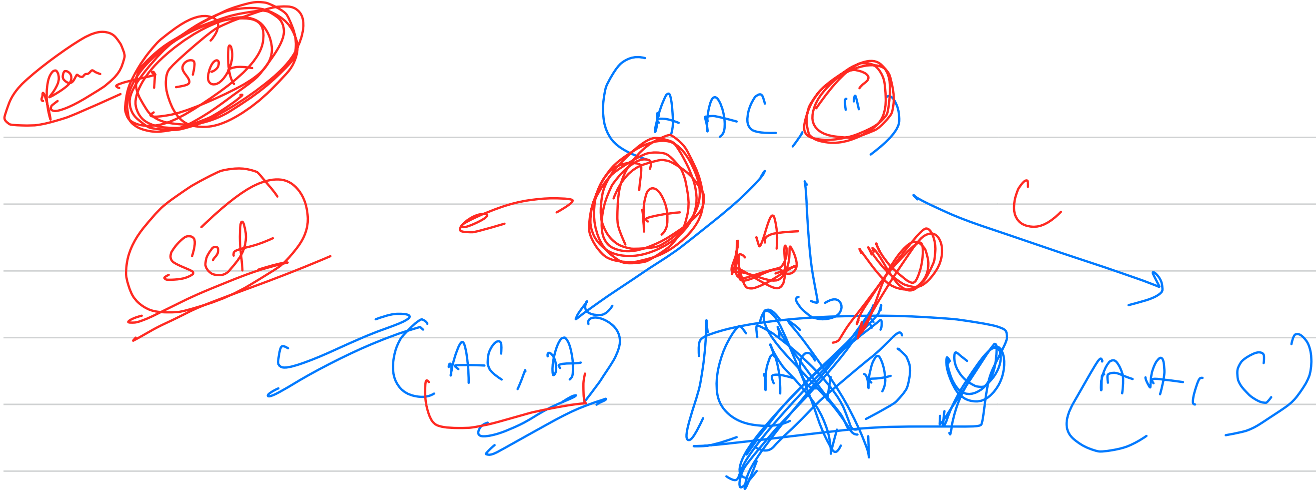
C++ / java / python   substring

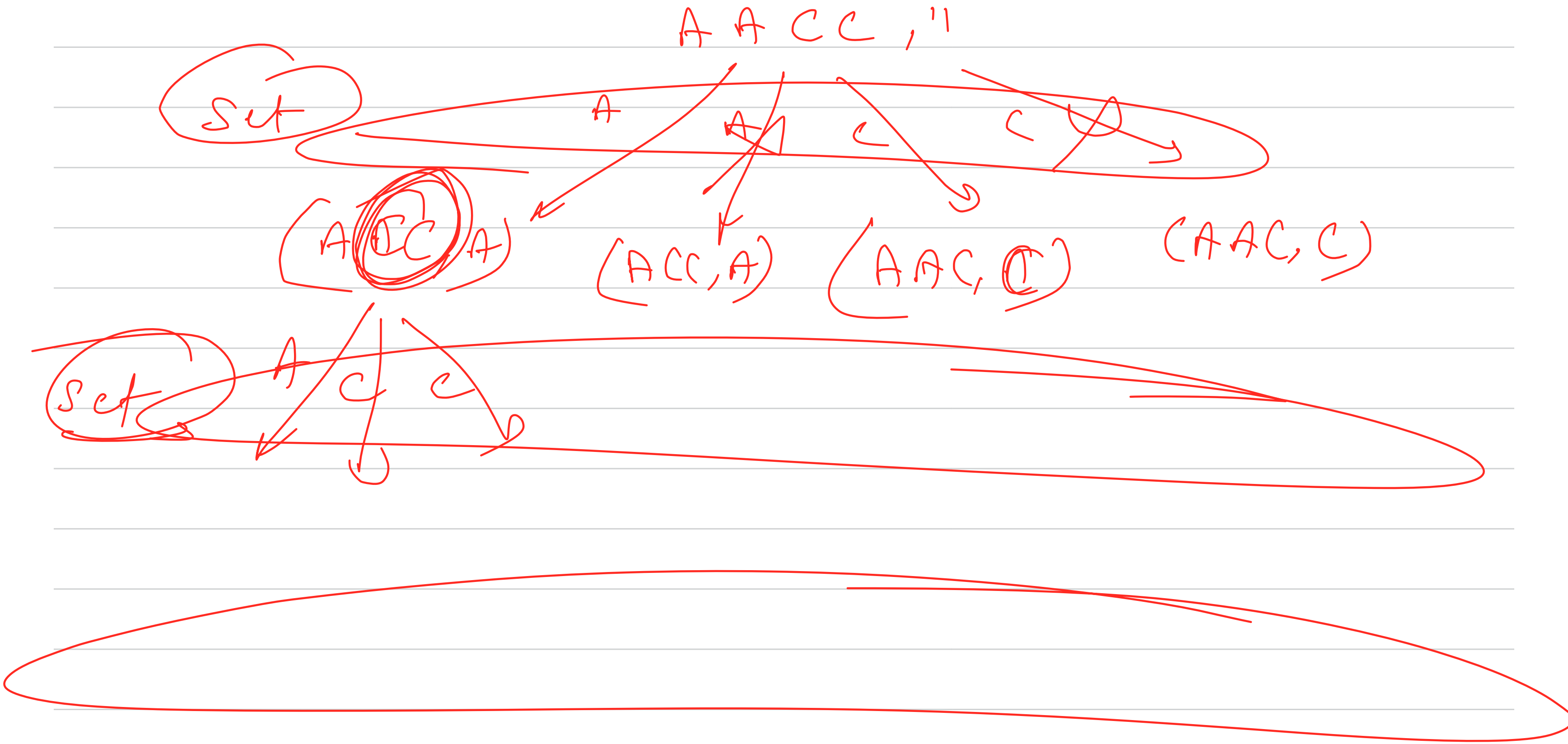
substr(idx, len) ⟶ substr(idx)

for (i = 0 ; i < str.size() ; i++) {

rest of string ⟵ ros = str.substr(0, i) + str.substr(i+1)

}


Set
Set
(AAC, "")
A
A
C
(AC, A)
(A, A)
(AA, C)



AACC, ''
set
A
A
C
C
(ACC, A)
(ACC, A)
(AAC, C)
(AAC, C)
set
A
C
C

(ABC, '')
(A, BC)
(BAC)
(CAB)