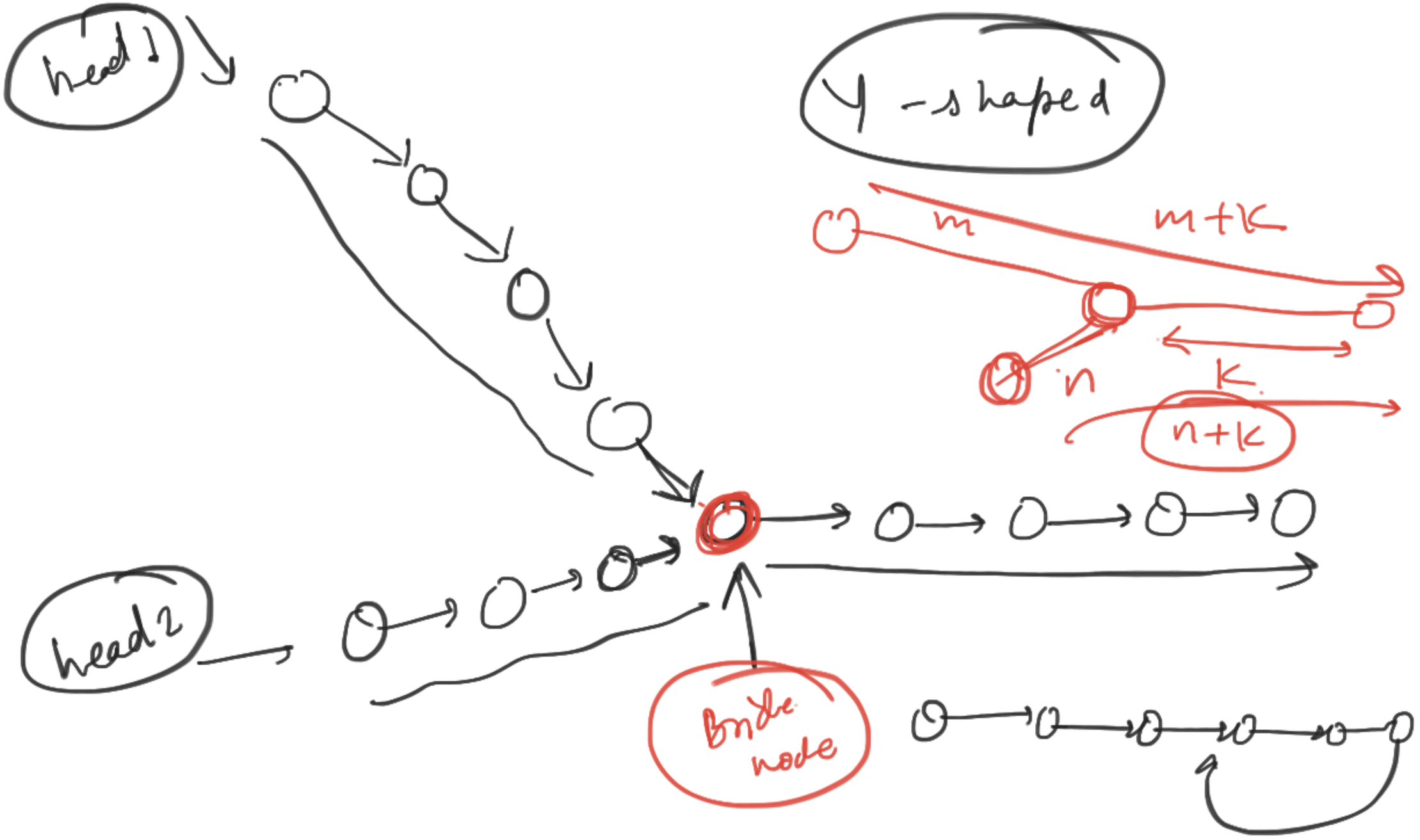
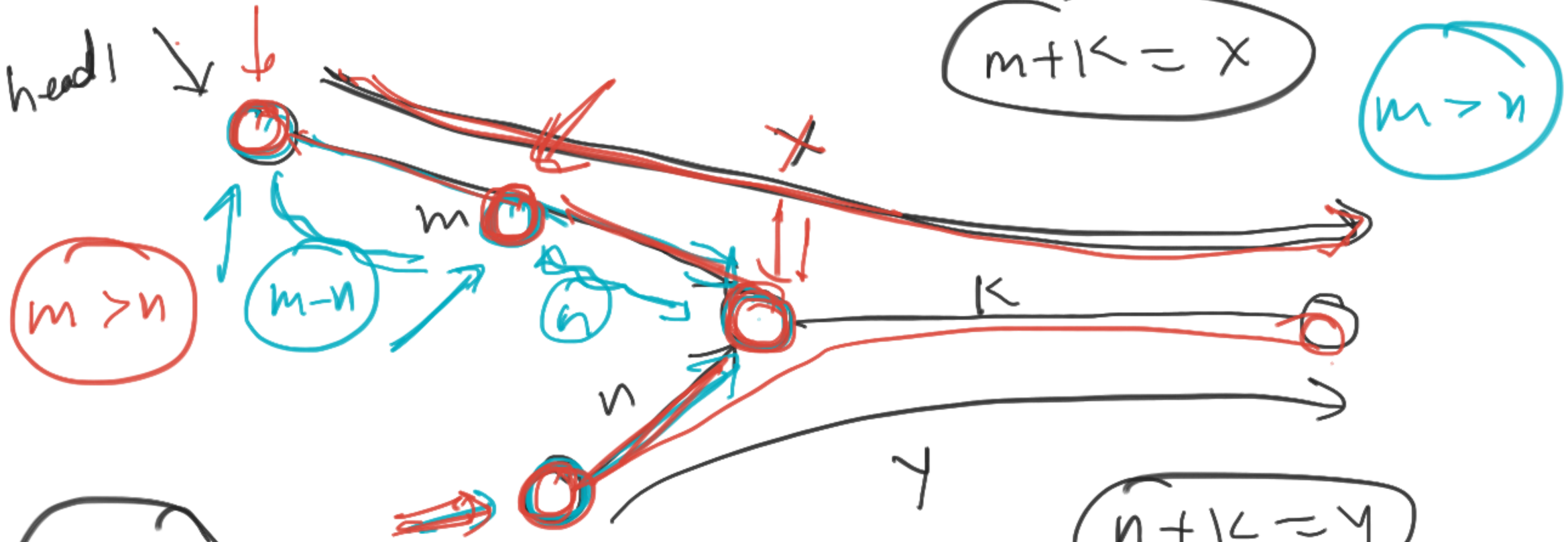

head1
Y-shaped
m
m+k
n
k
n+k
head2
Bride node



m + k

n + k

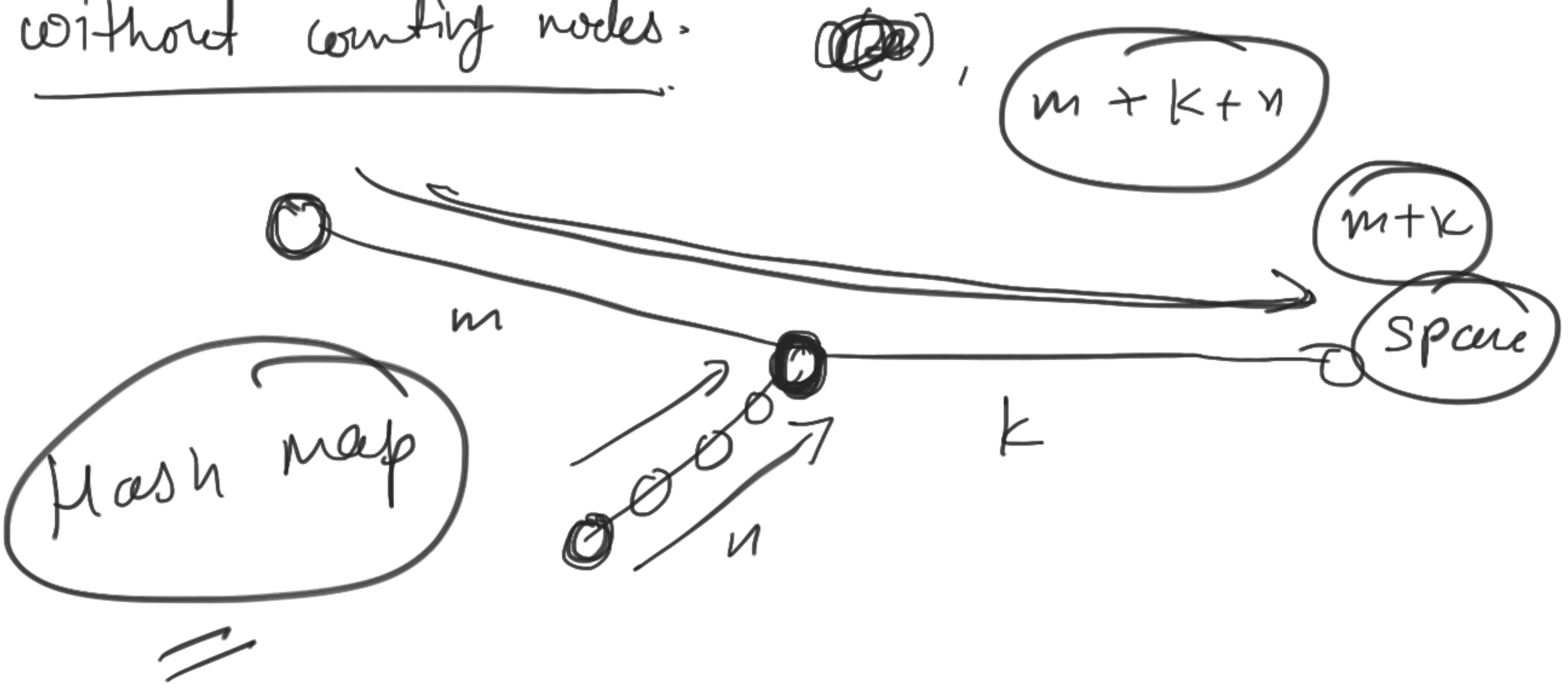
m - n

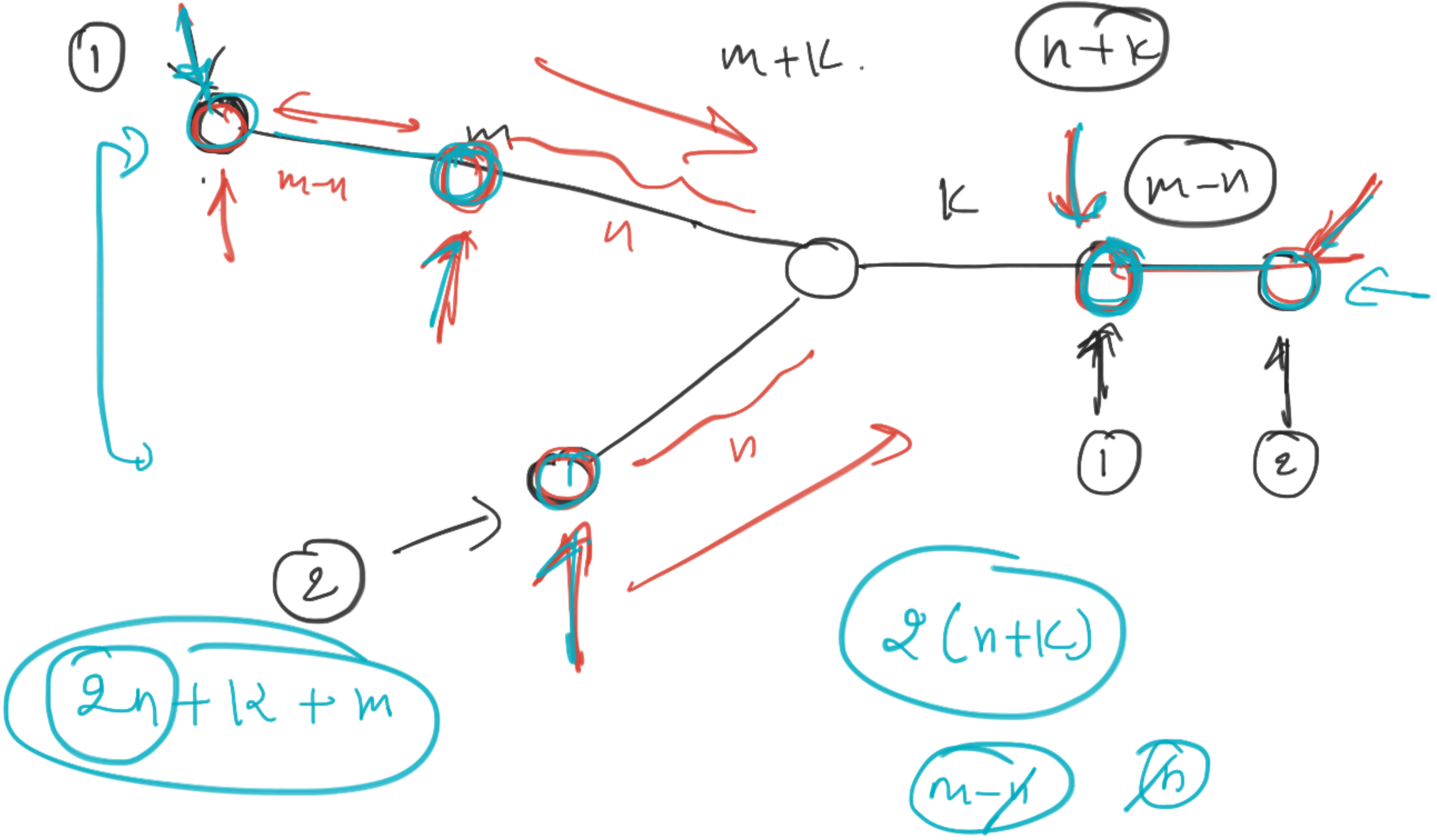
$\Rightarrow$ 2m + 2k + n

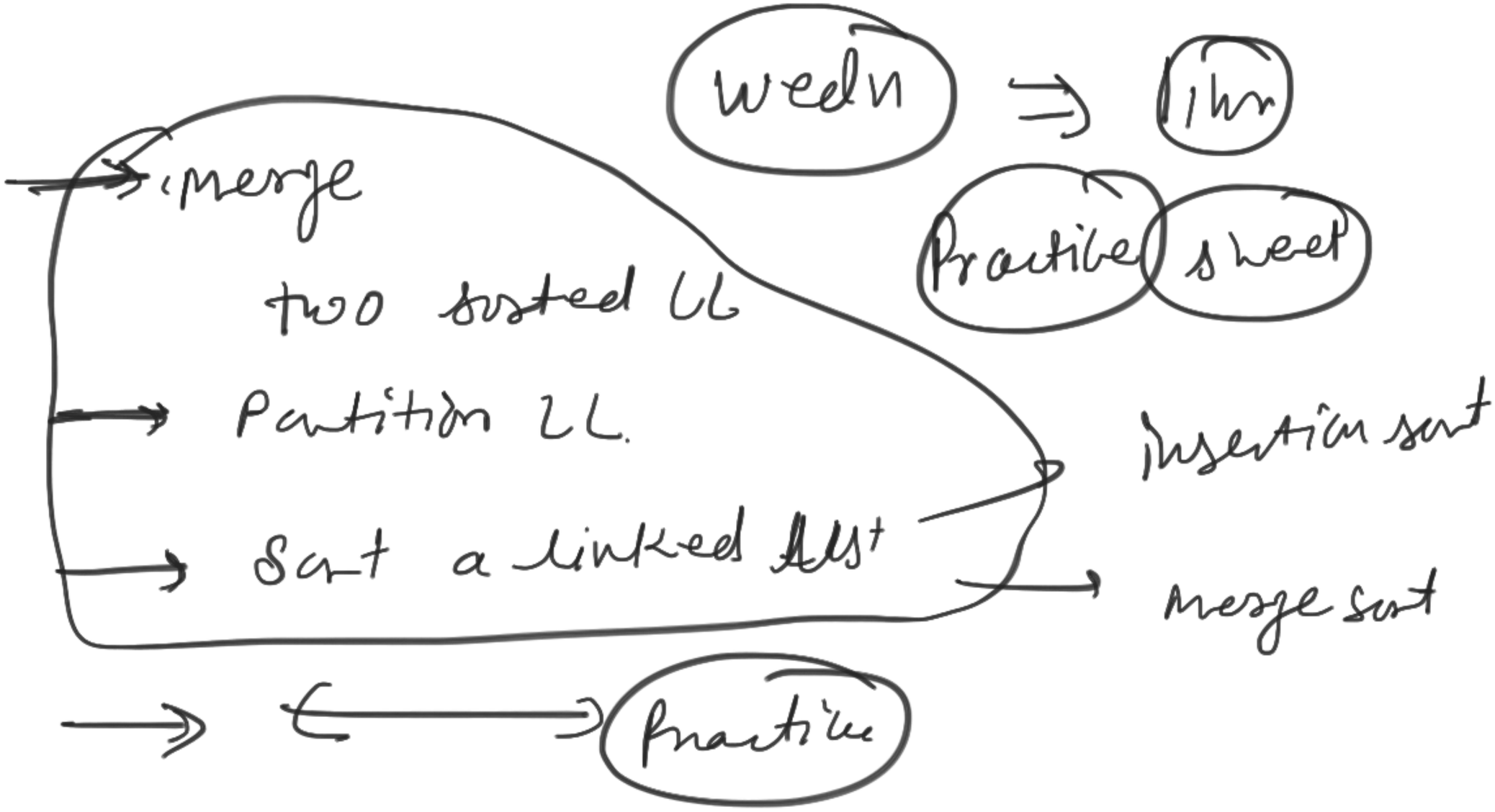
x - y = (m+k) - (n+k)

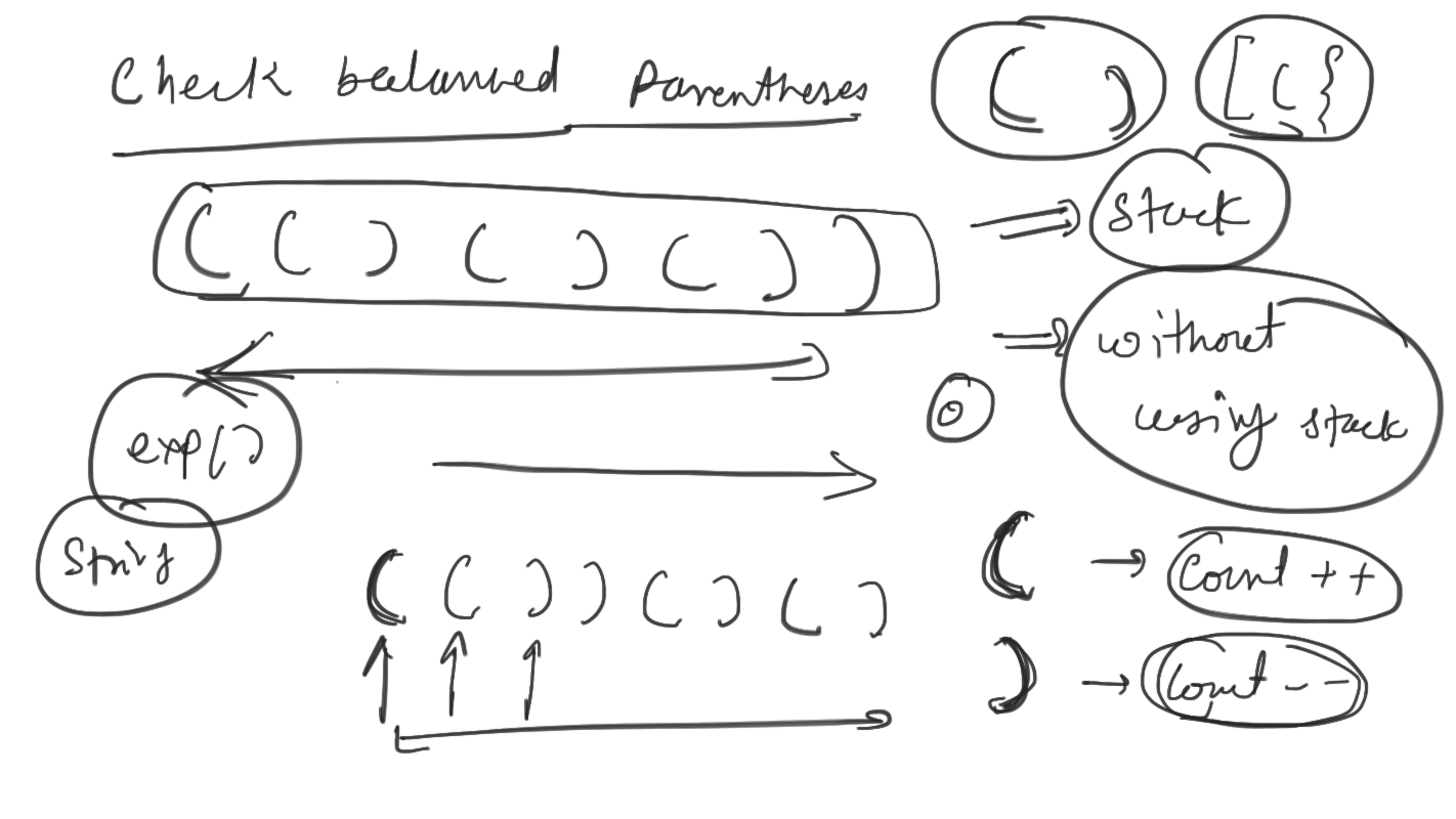
= m - n

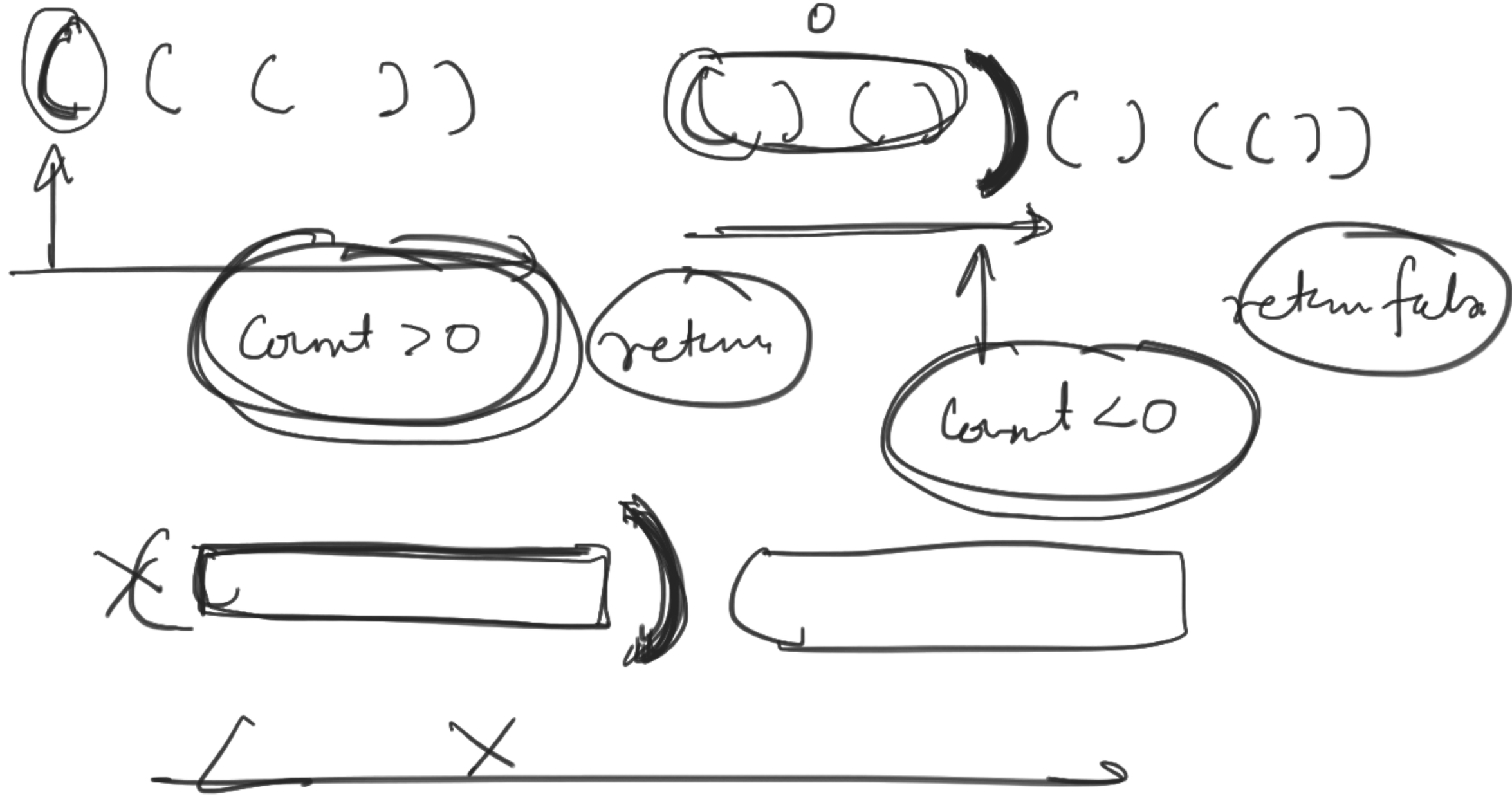
without counting nodes.

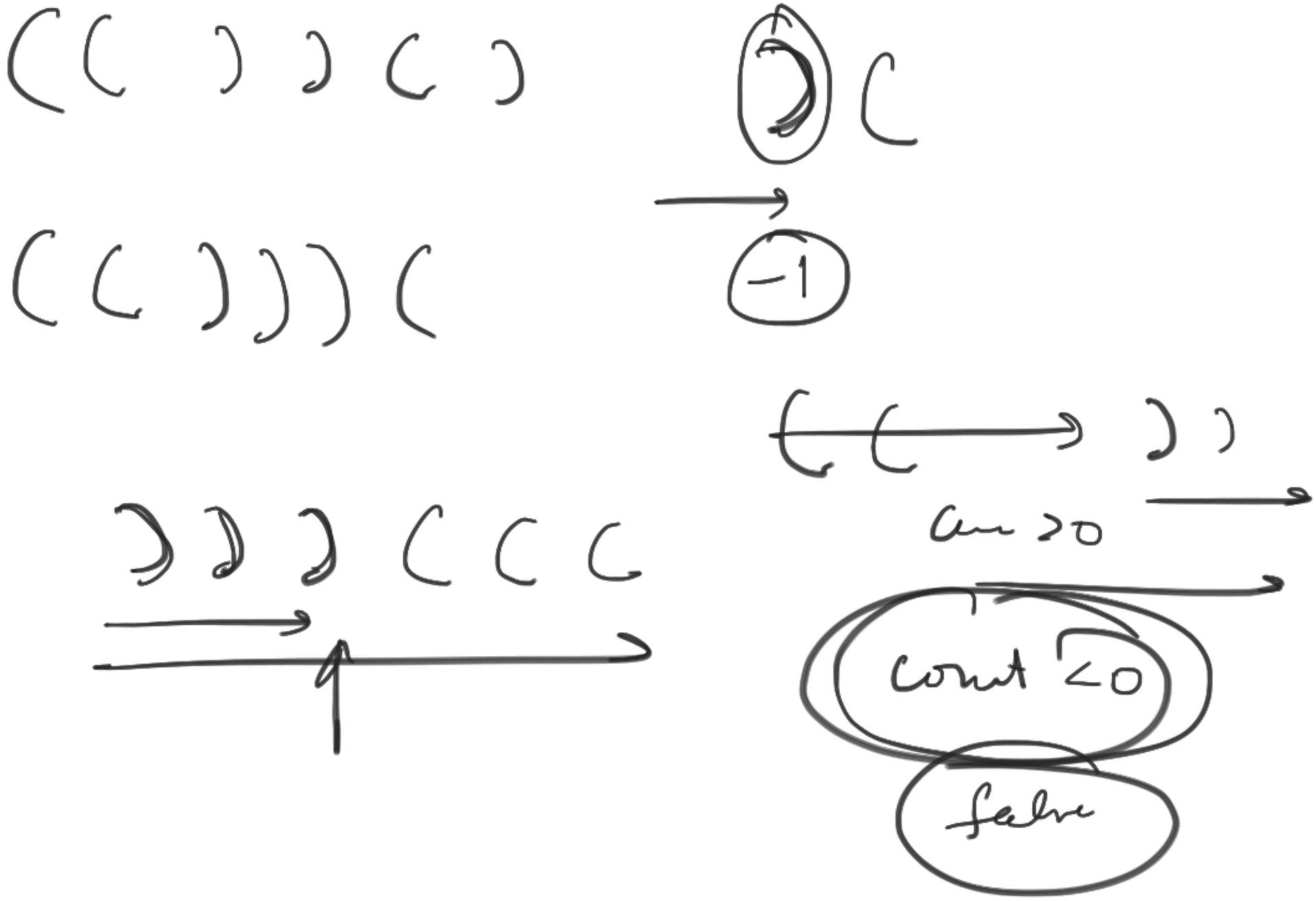



1
m+k.
n+k
m
m-n
n
k
m-n
n
1
2
2
2n+k+m
2(n+k)
m-n
b


weeln
1hr
Practice
sweep
merge
two sorted LL
Partition LL.
Sort a linked list
insertion sort
merge sort
practice


Check balanced Parentheses
( ) [{ }
Stack
(( ) ( ) ( ))
without using stack
exp( )
String
0
(→ Count ++
) → Count --


0
(( ))
( ) (( ))
Count > 0
return
Count < 0
return false


count >0
count <0
false

for(i = 0 to n-1)

{ if ('(')

count++

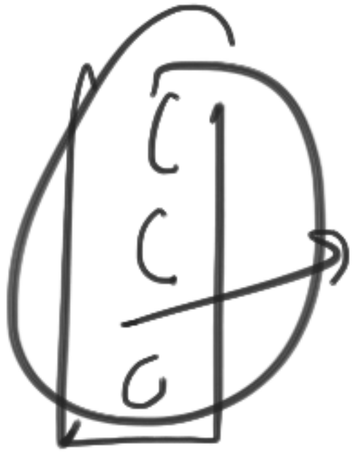
else

count--

if (count < 0)

return false.

}



if (count != 0)

return false

extra opening

return true

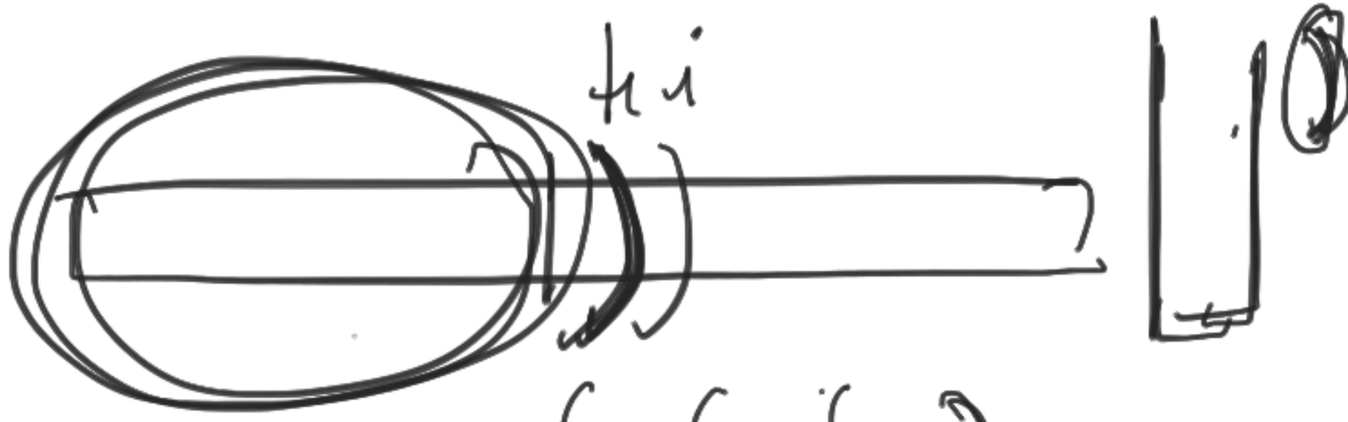
count > 0

} min no. of removal to make it balanced

extra closing

extraOpen = 0

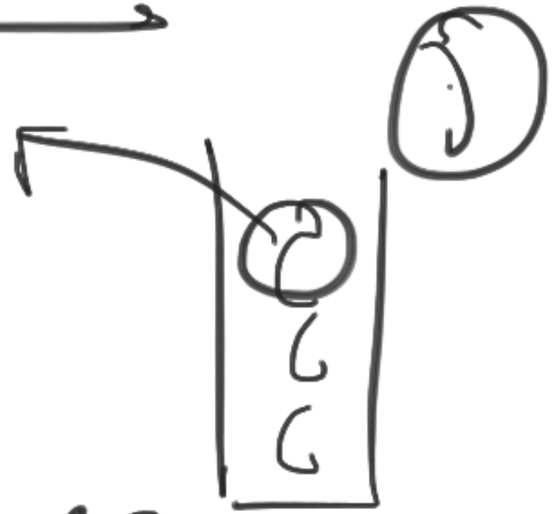
extraClose = 0



for (i = 0 to n-1)

{ if (s[i] = '(')

extraOpen = extraOpen + 1

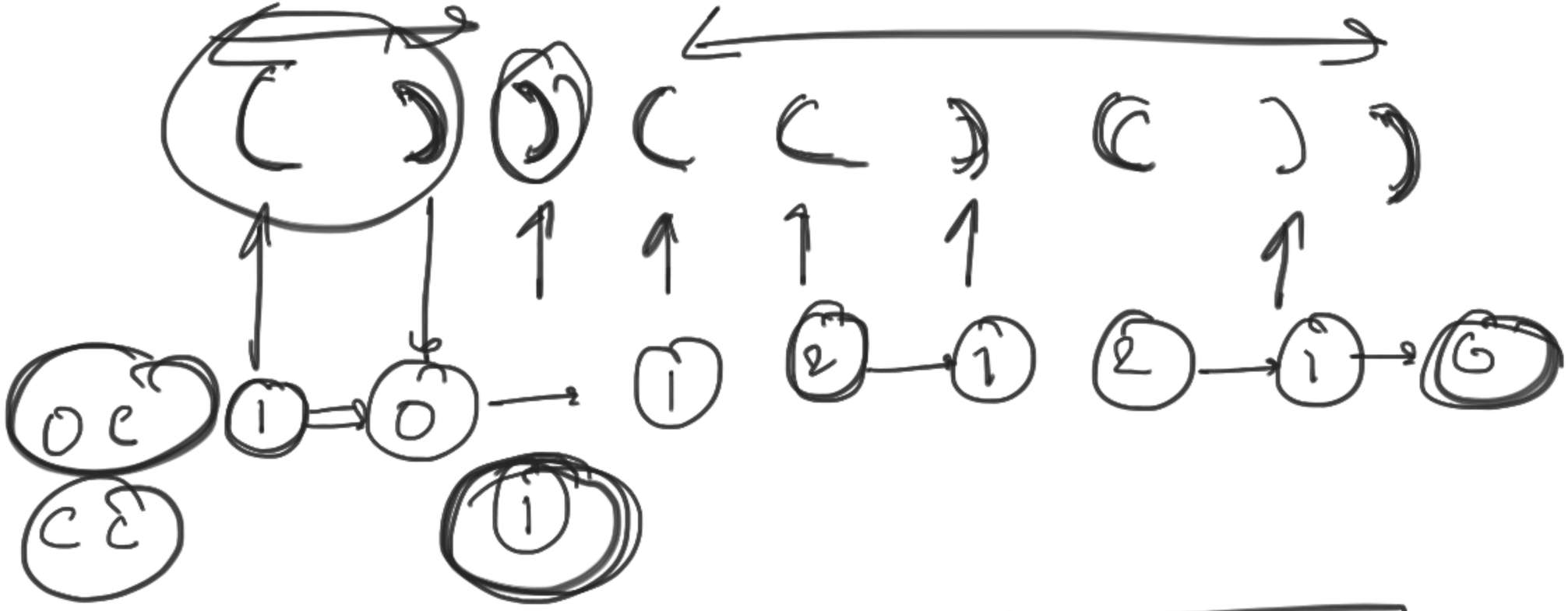


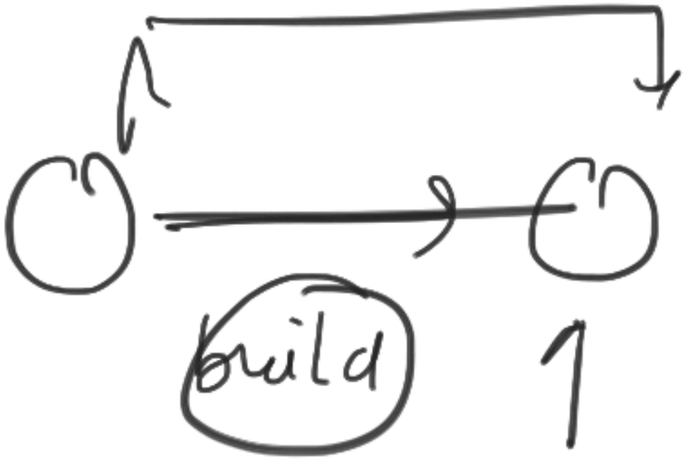
else { if (extraOpen > 0 && s[i] ==))

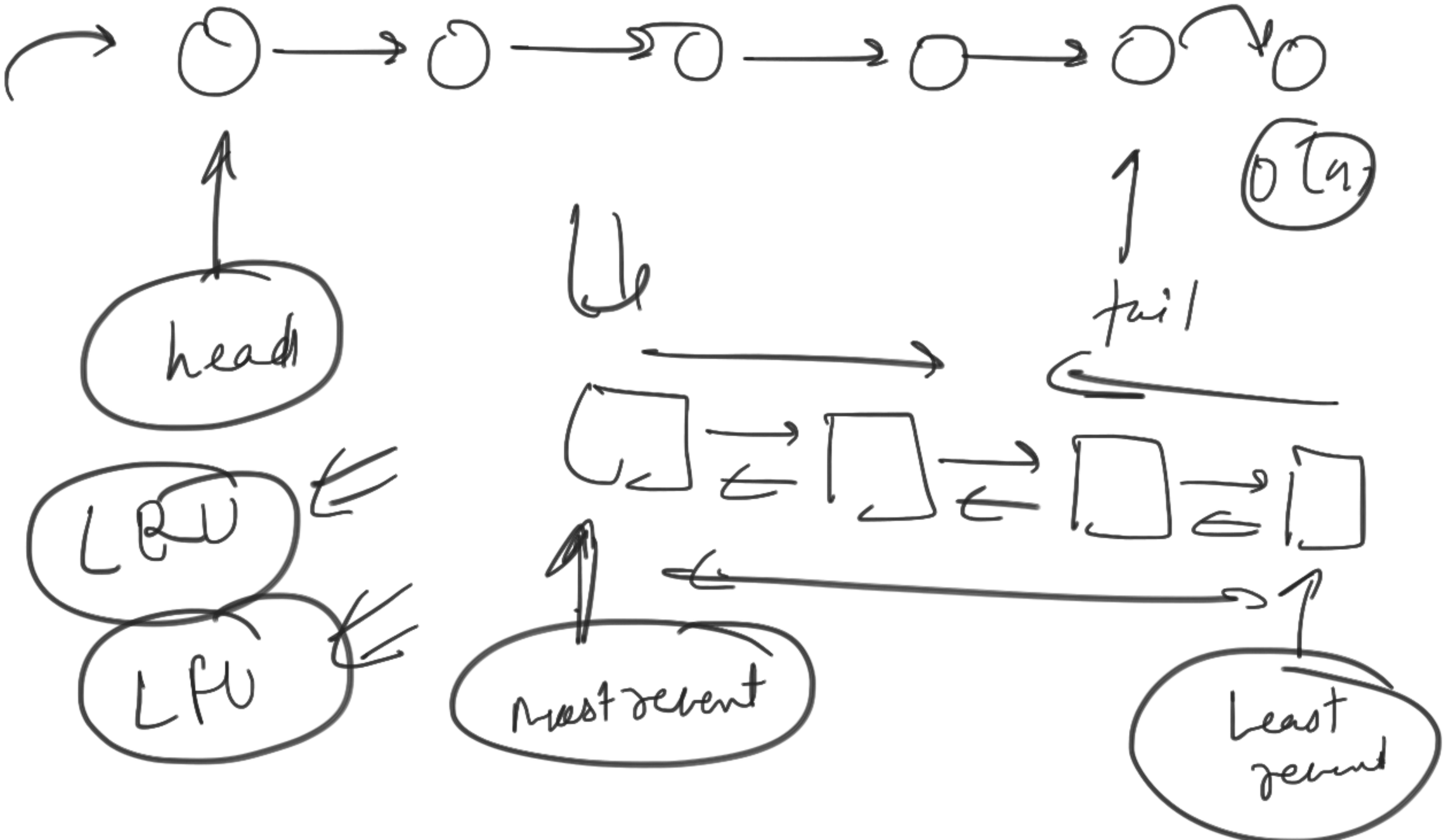
extraOpen --

} else extraClose ++

return expression + expr.


0
1
0
1
2
1
2
1
0


build
1


head
LRU
LFU
tail
Most recent
Least recent