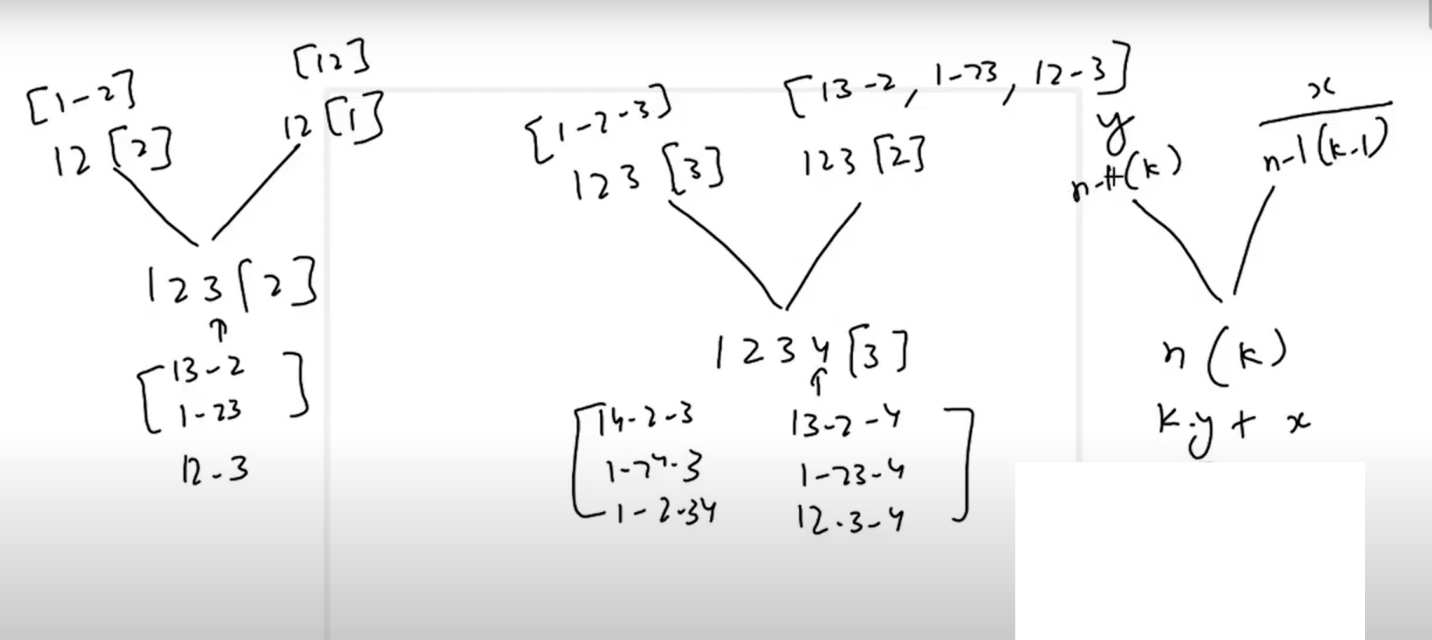
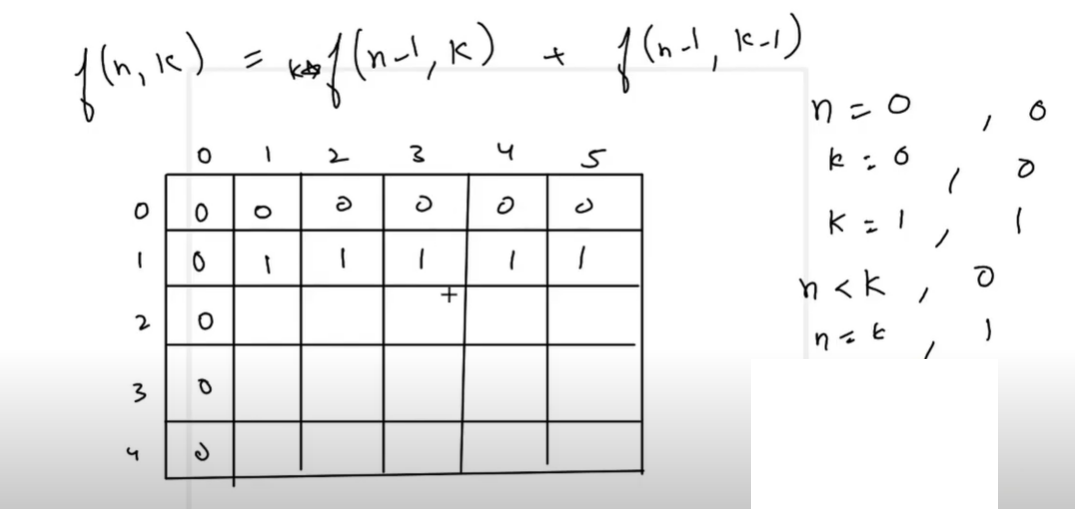
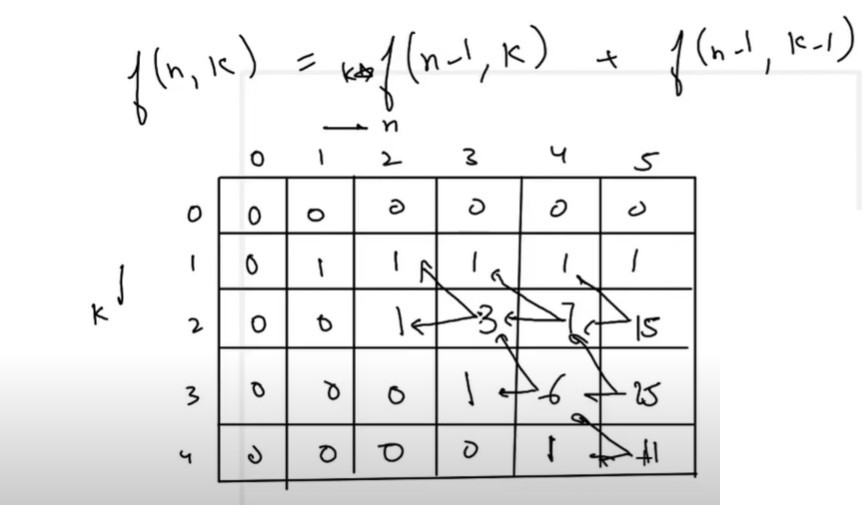
N people making K team with combination.

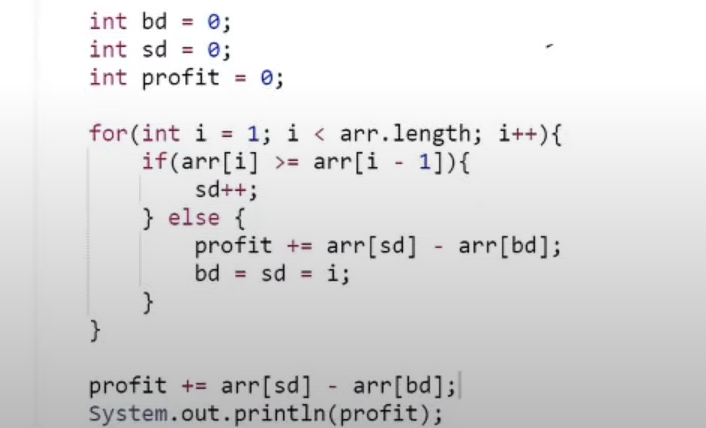


F(n,k) = k \* f(n-1, k) + f(n-1,k-1)



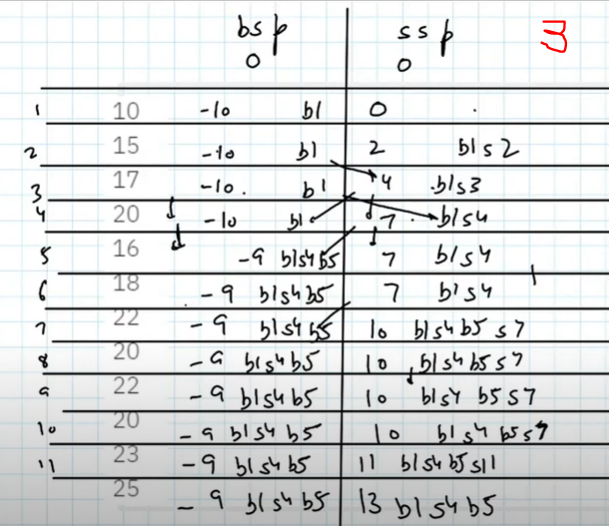


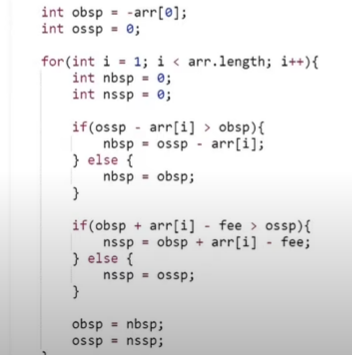
Buy sell stock with unlimited transaction. Logic is increase selling date if it is greater then previous value and if is smaller than previous then do transcation and them move both buying date/selling date to current. NOTE: Last upstrock is not covered in loop because there is not dip in graph so we need to handle out side of loop



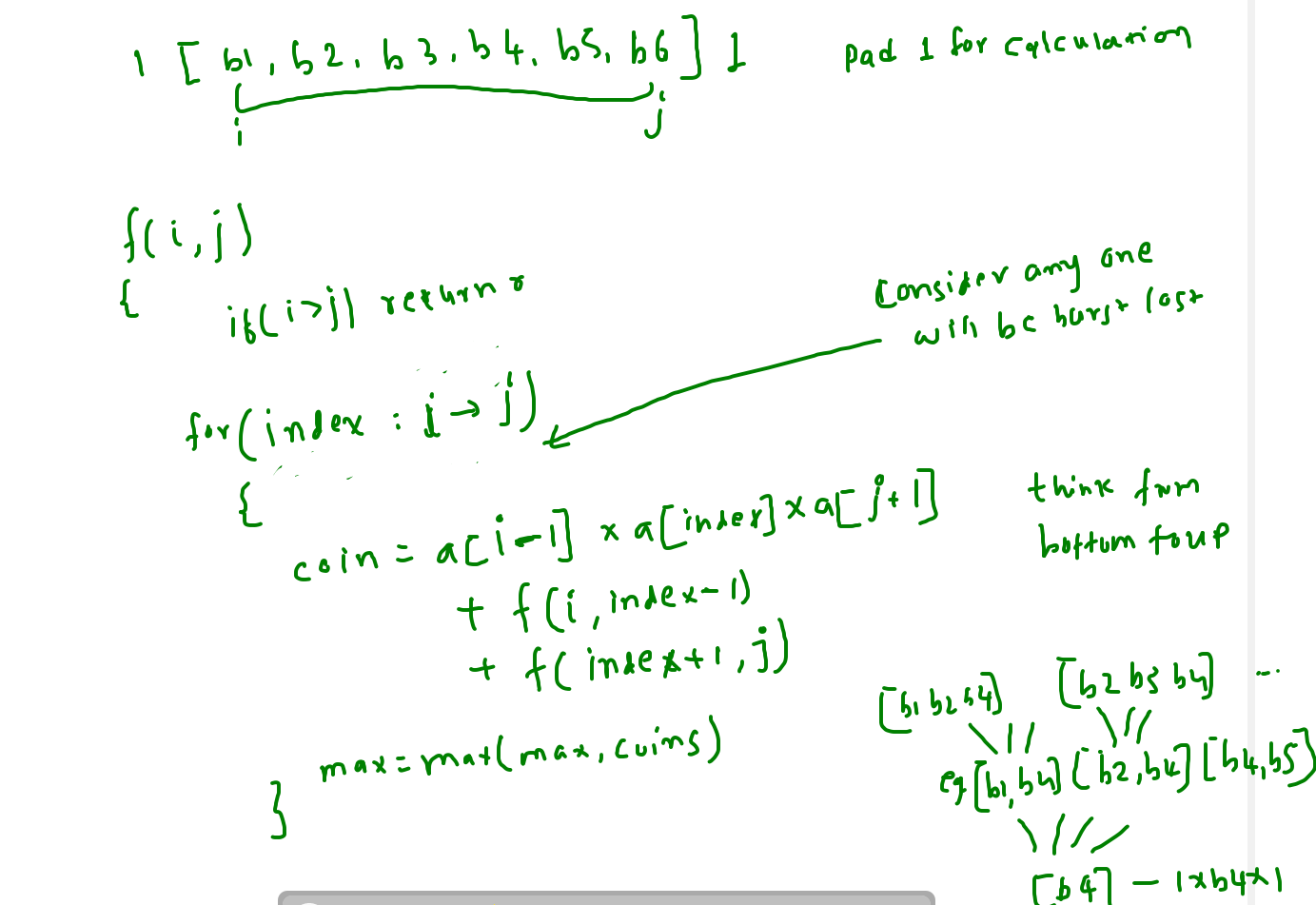
Buy sell stock with fee and unlimited transaction.

Include exclude pattern.

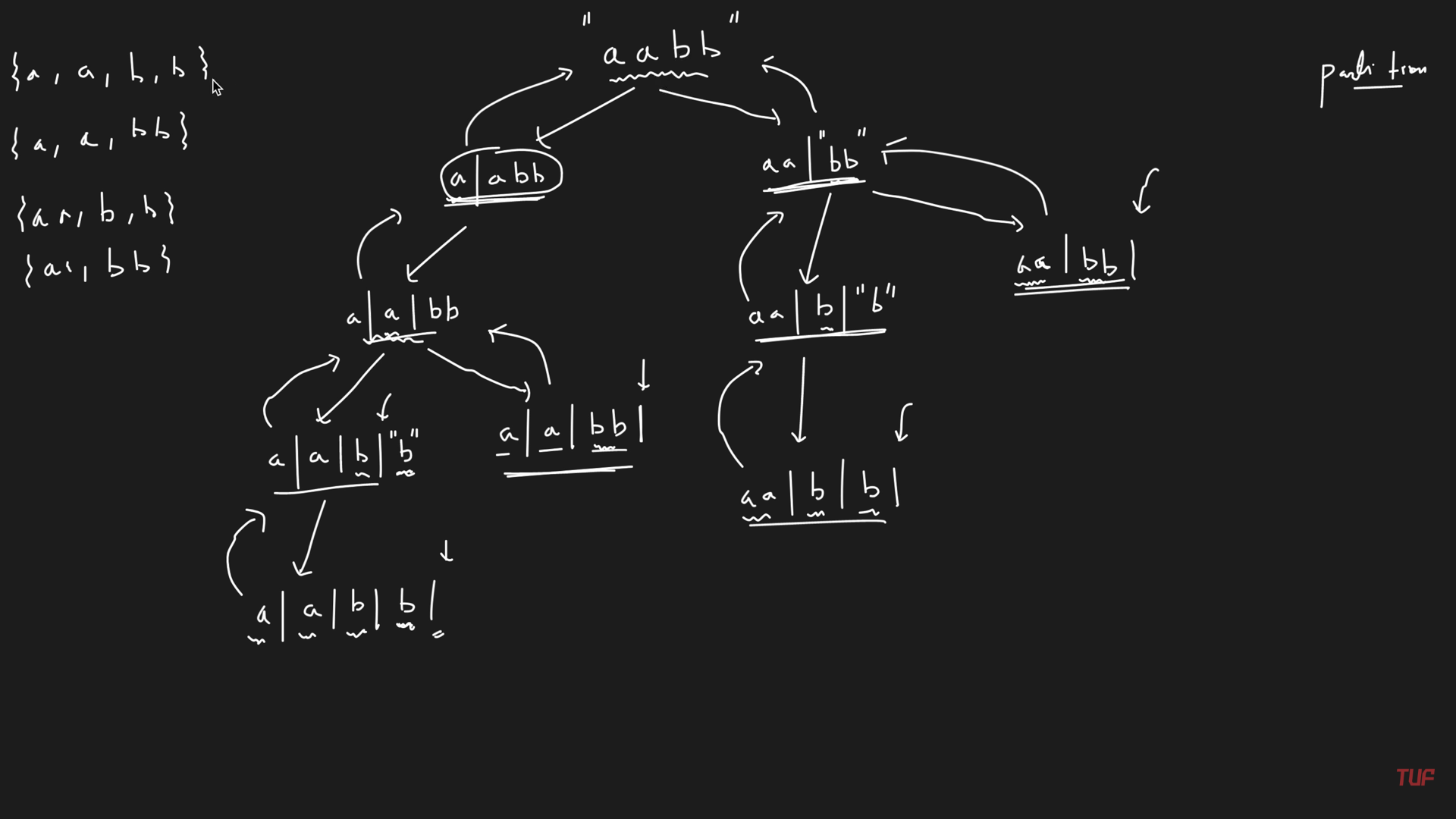


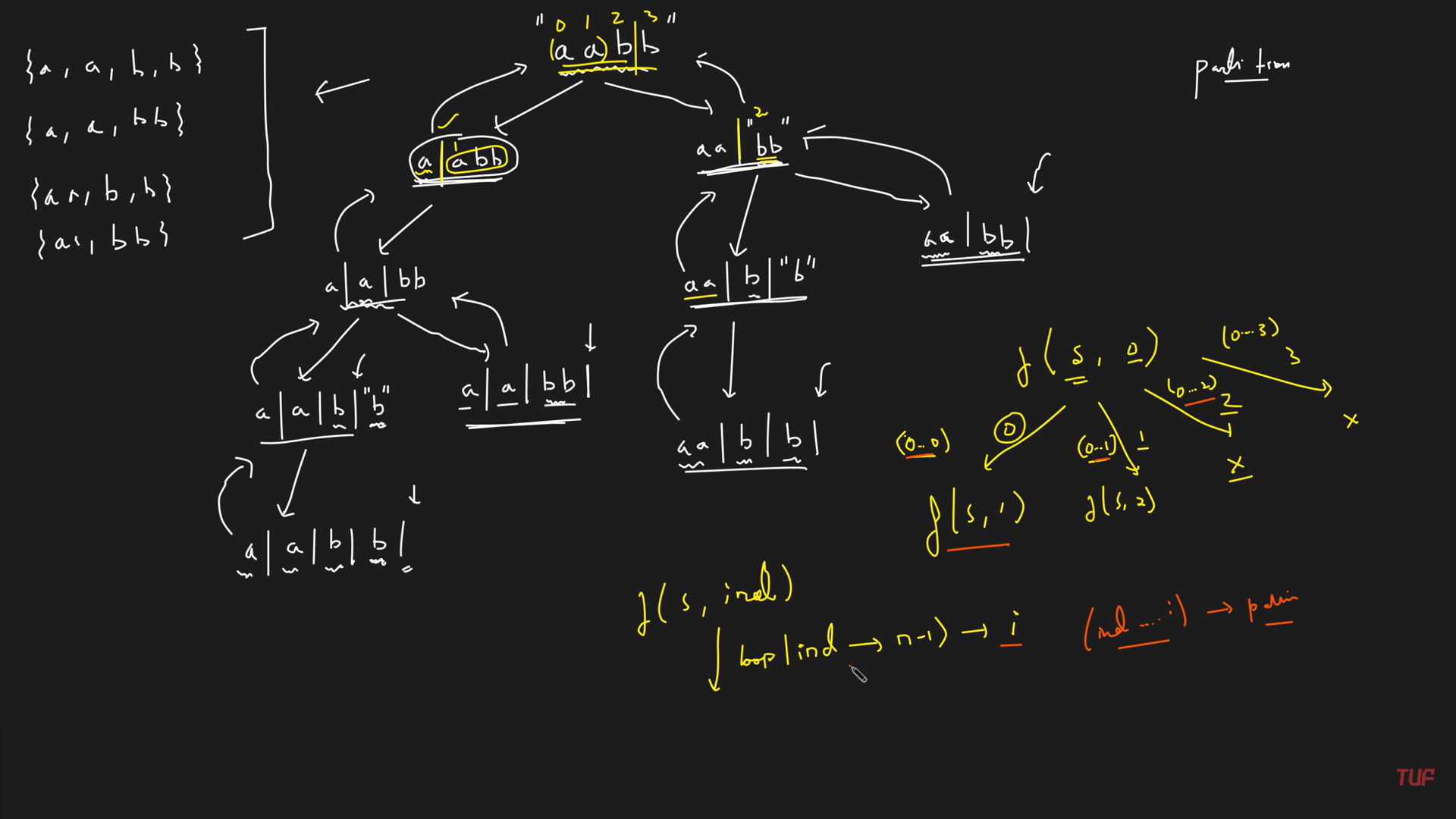


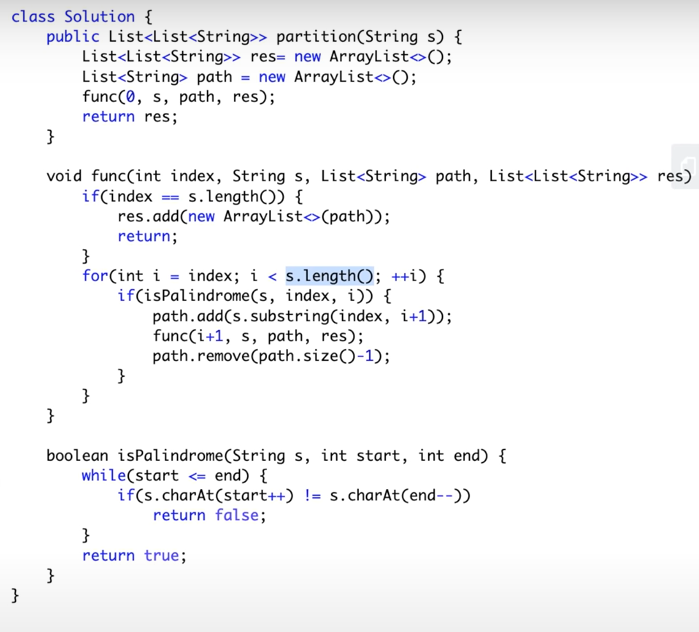
Burst Balloon problem

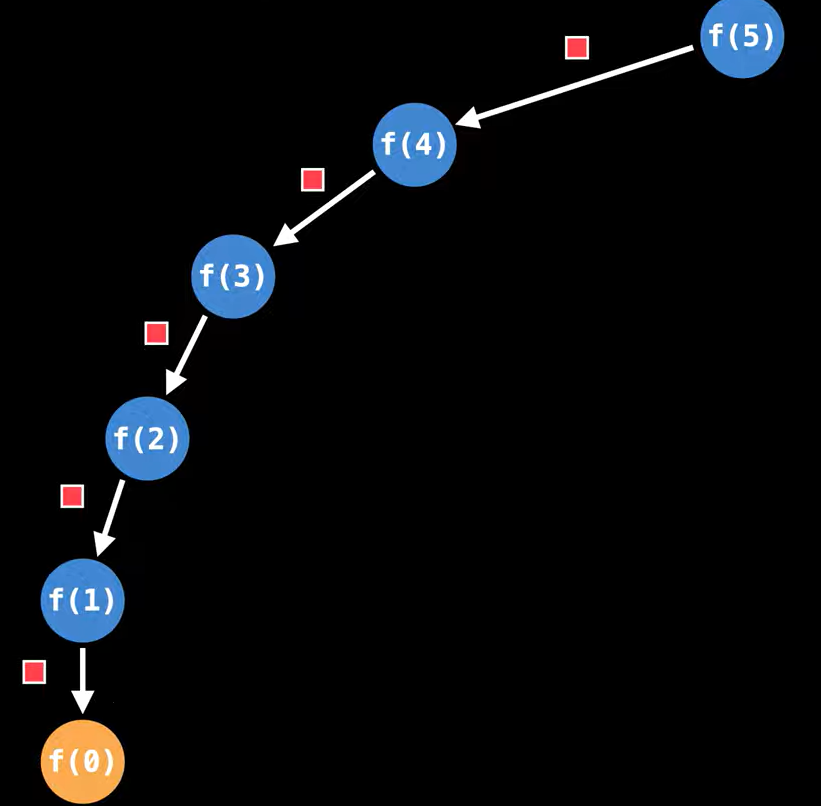


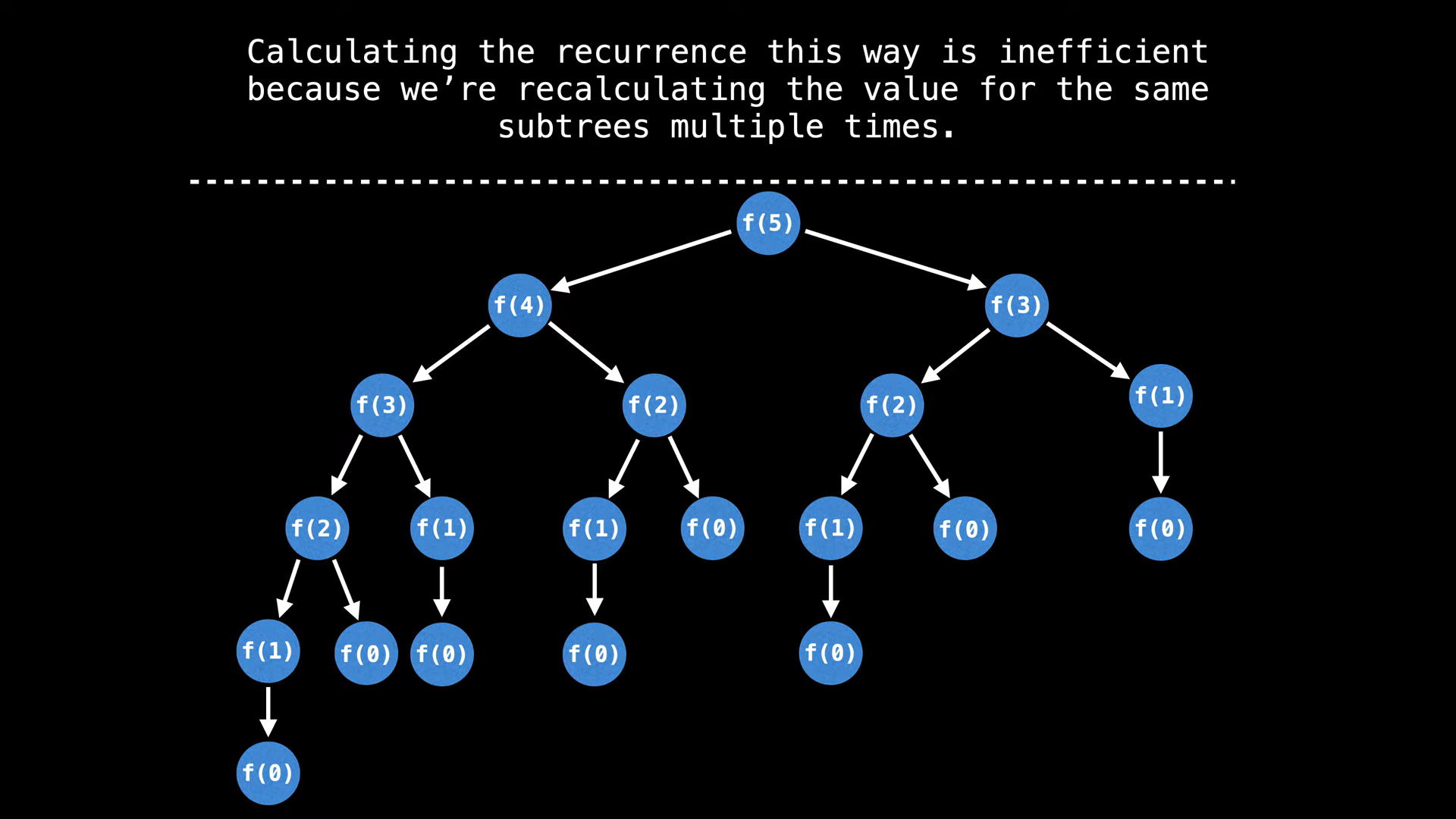
Palindrom partition

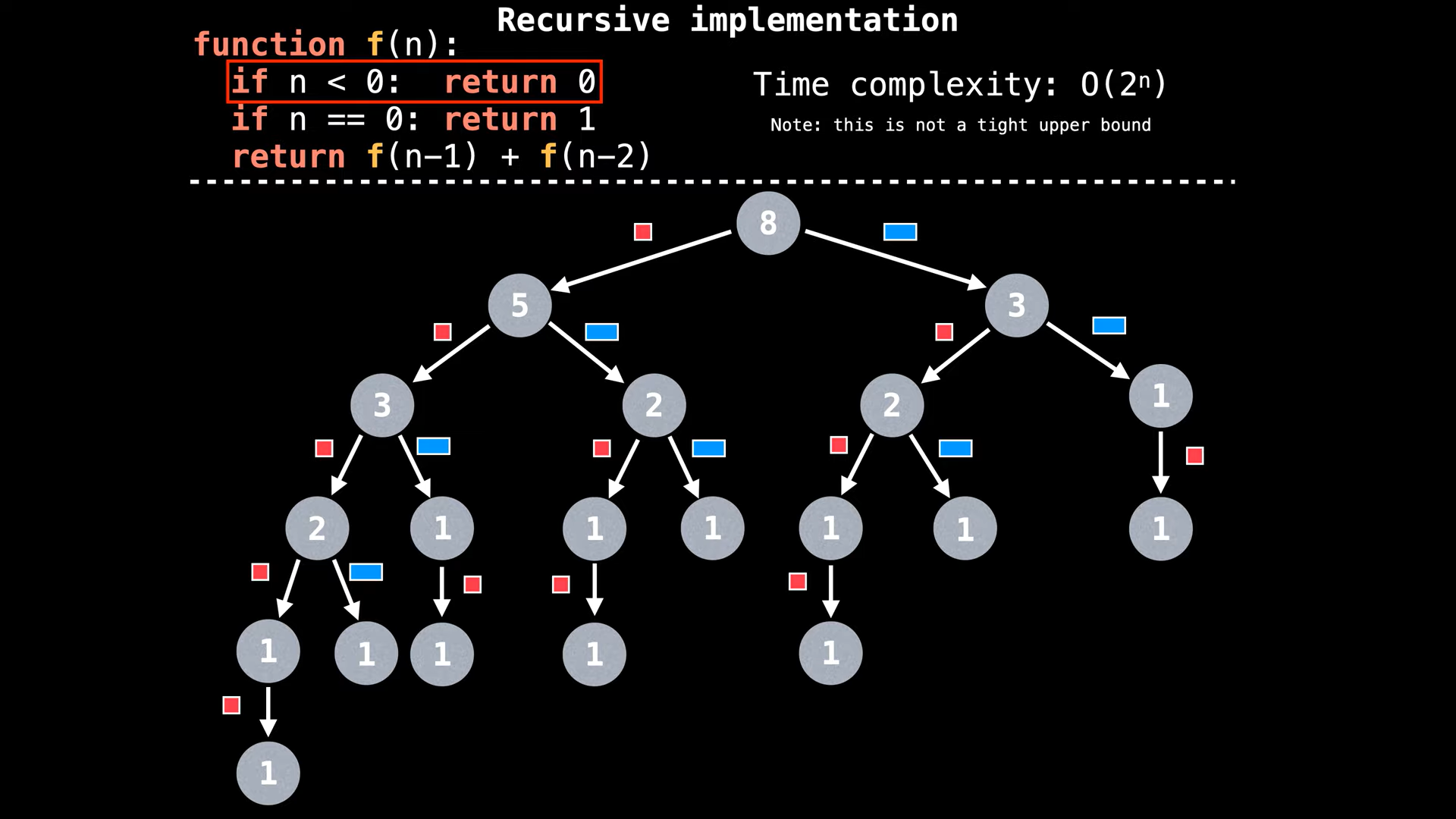




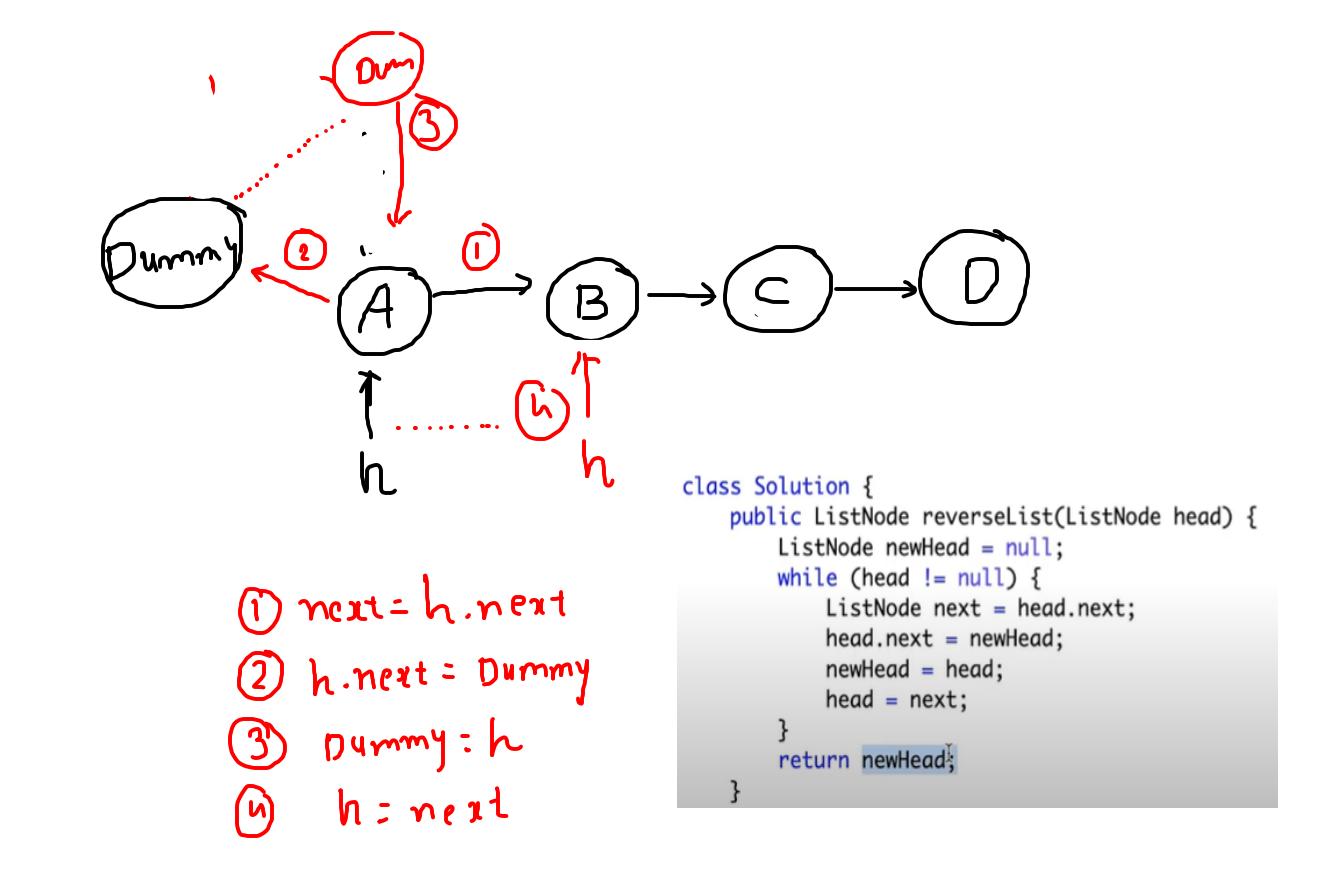


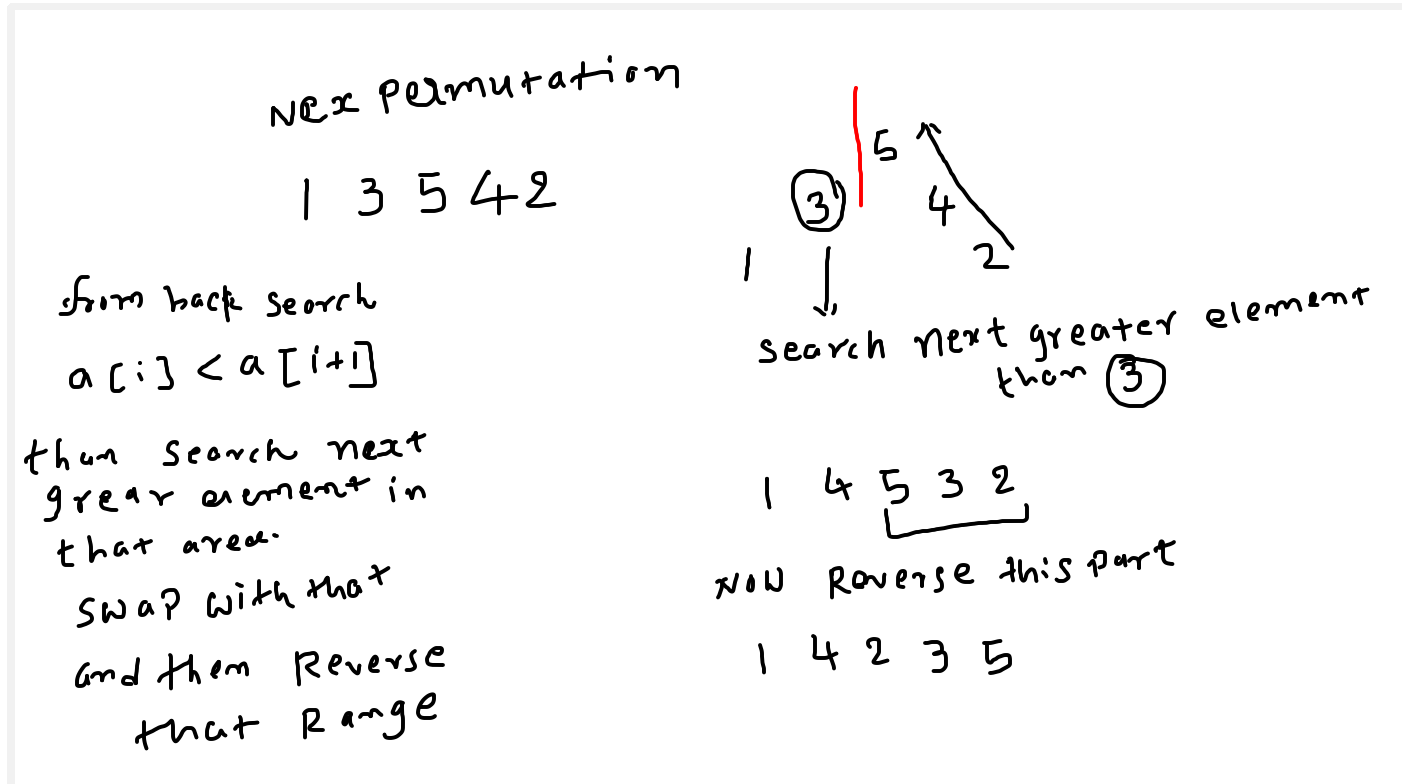






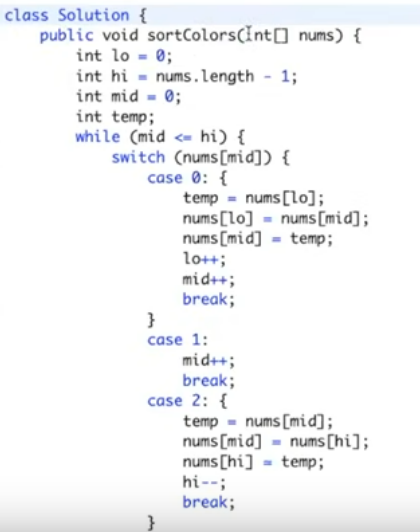
Reverse Linked List





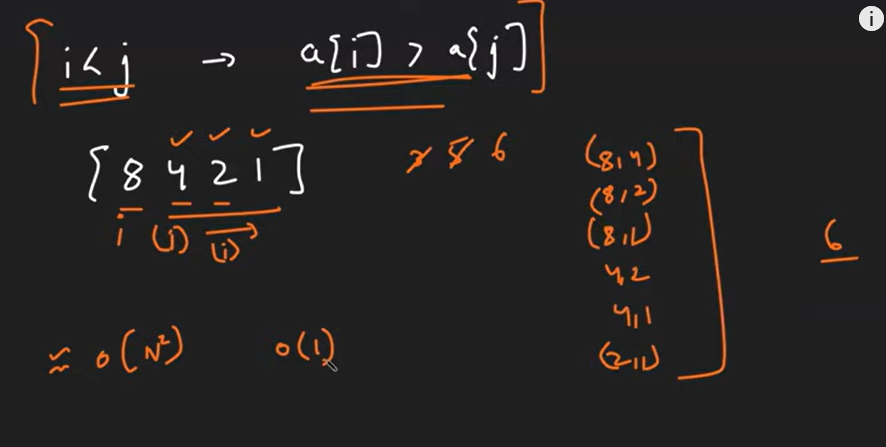
Sort array which has only 0 , 1 , 2 element. Use dutch flag algoright.

Keep 3 pointer low , mid, high..

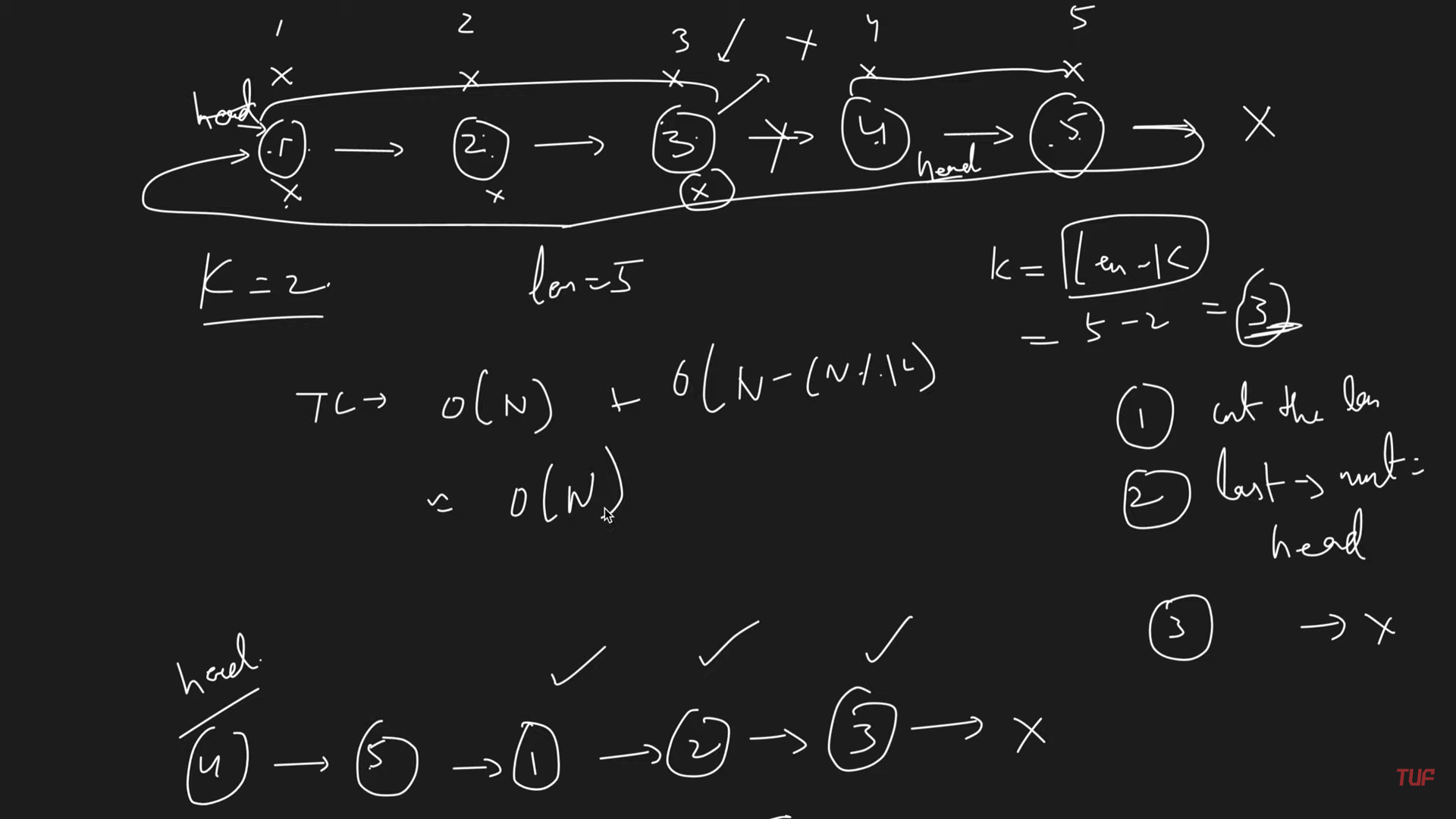
All the zero will be left of low, and all the 2s will be right of high and all the ones will be in mid area.

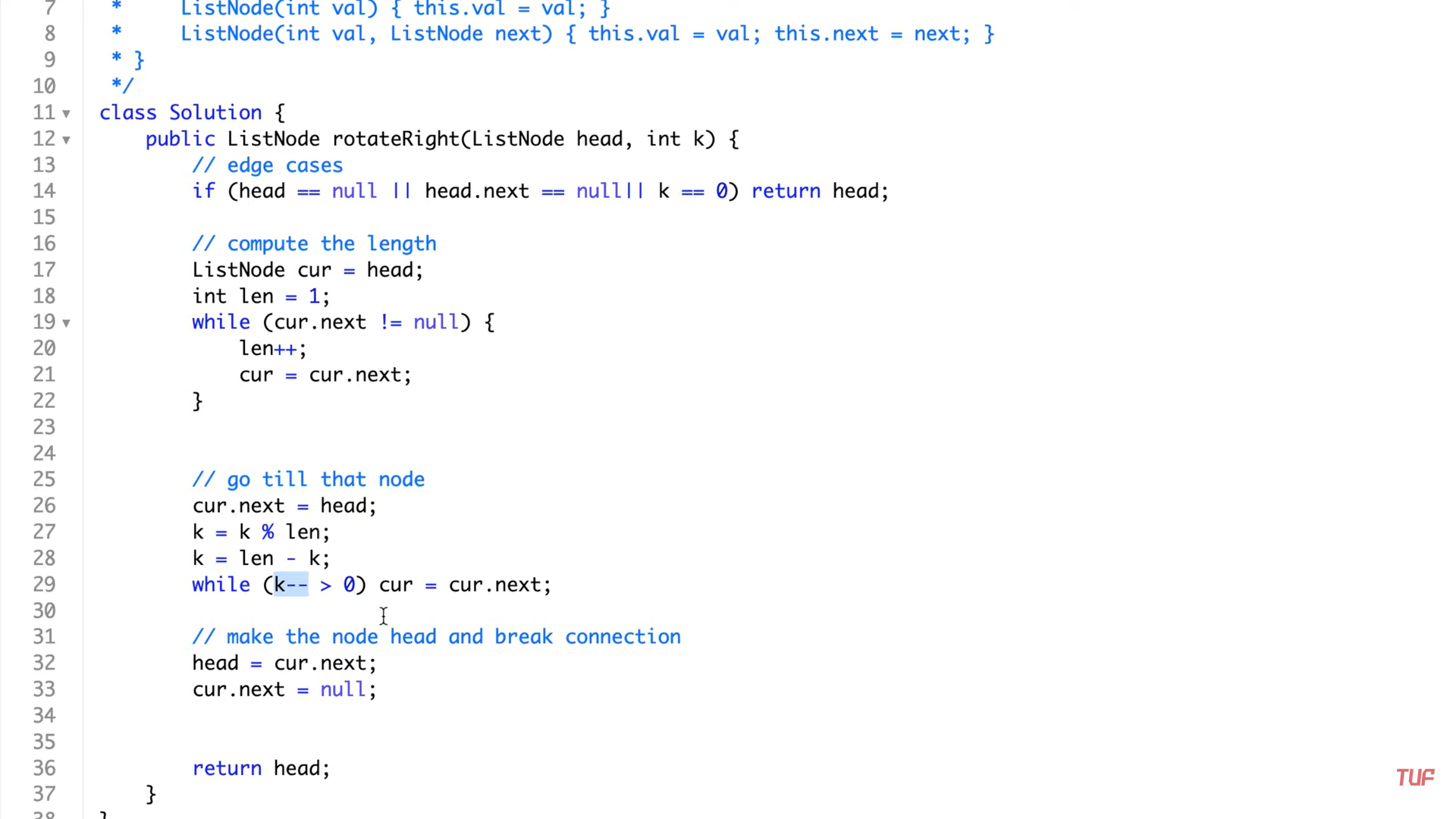
**COUNT INVERSIONS in an ARRAY**

Use the merge sort and count the variable which falls under this condition.



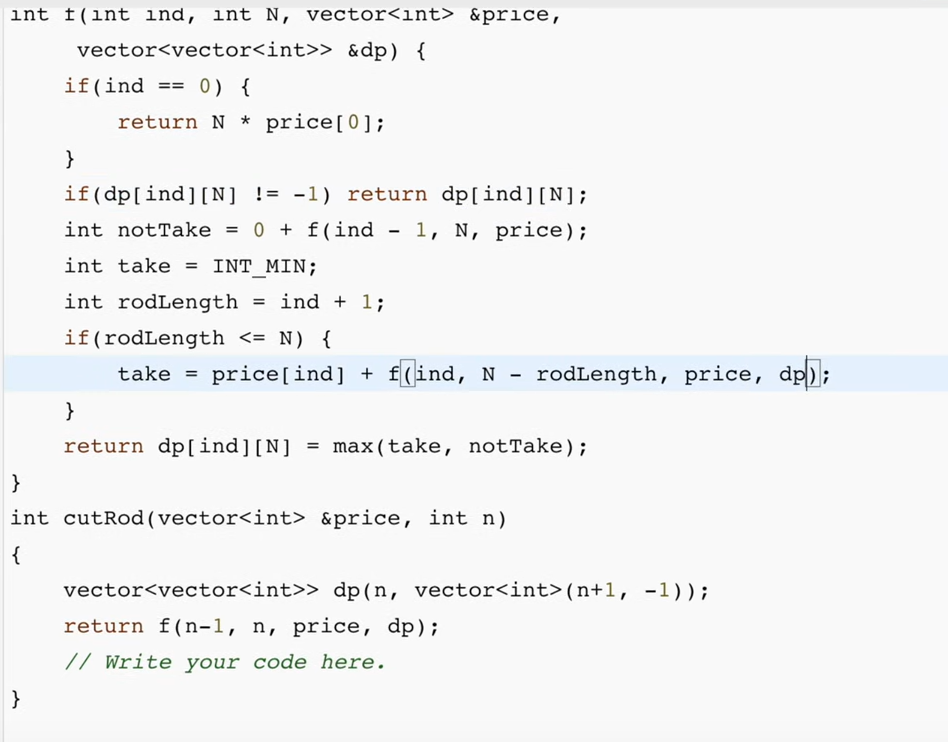
Rotate Linked list

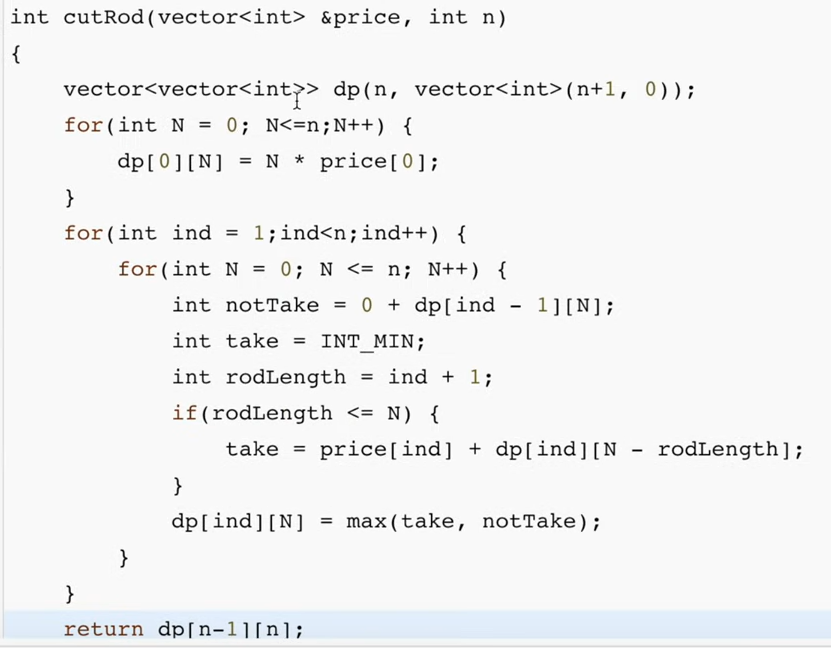




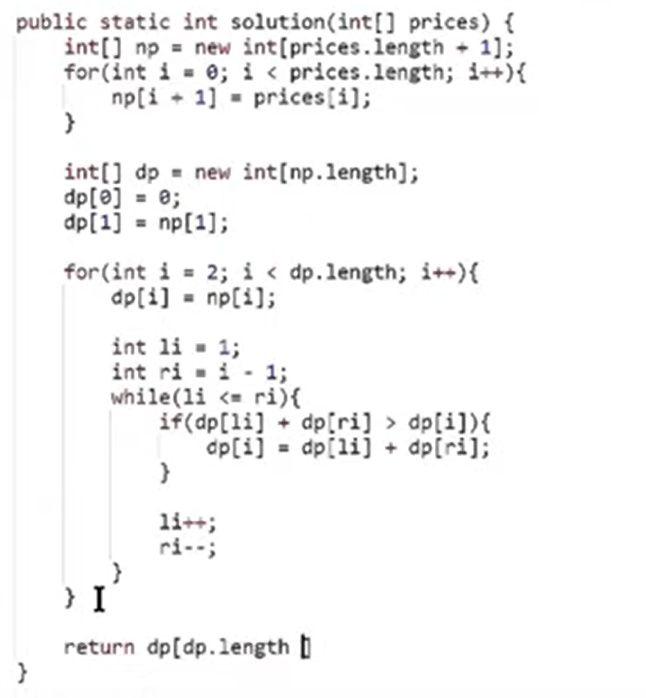
ROD CUTTING max profit

Approach #1 (Unbounded knapsack)

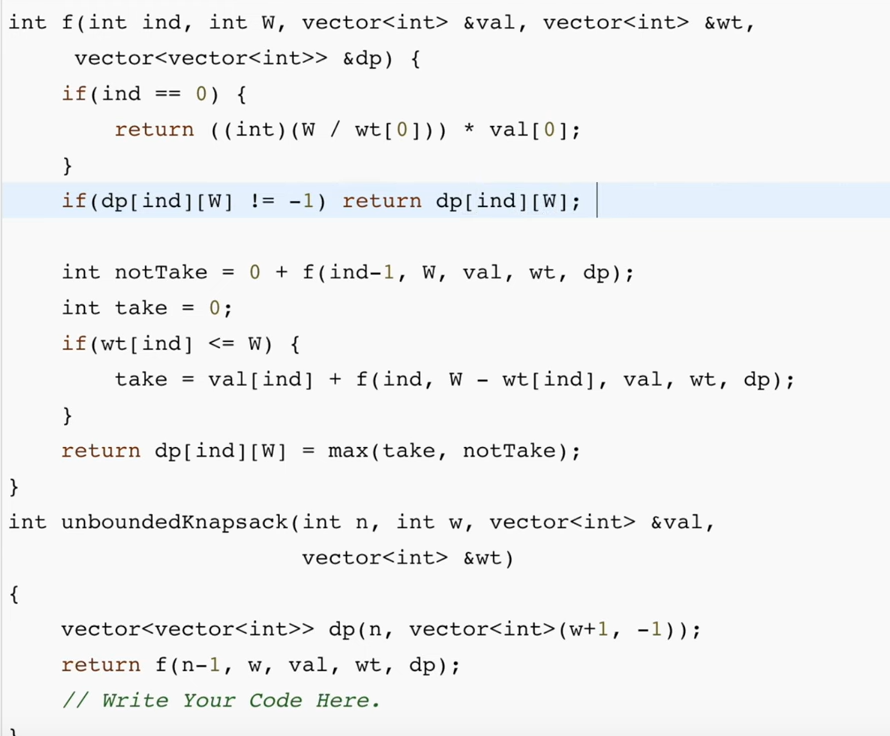


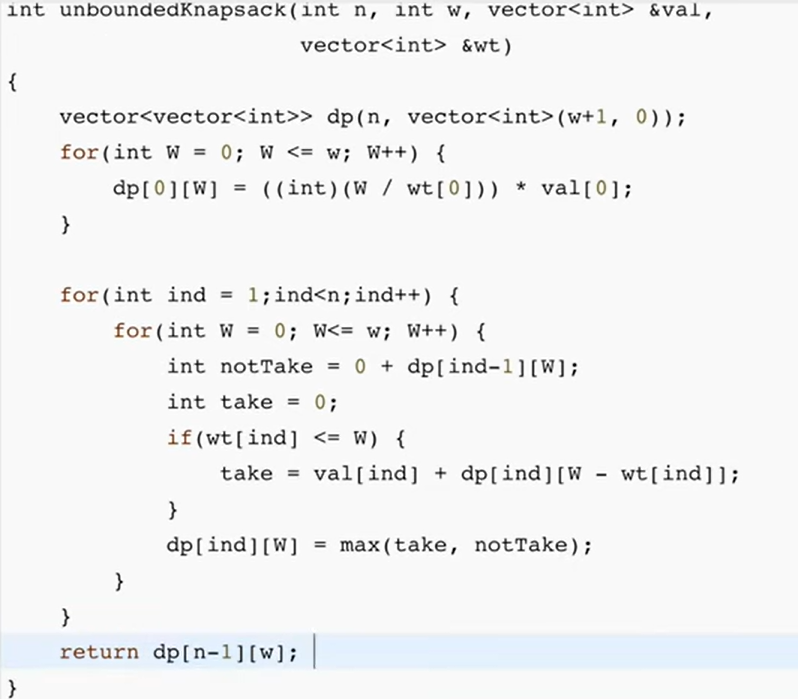


Approach #2 (Left/Right stategy)



Unbounded Knapsack:





MinStack with O(1)

First item will be in min=item1

If (newitme < min)

Newitem = 2 \* newitem – min; and store this new

Min = newmin

Removal

If popitme<min

Popitem = 2 \* min – popitem;

BIT BINARY

**Logbasenumber** No of digit in any base number digit = (int) ( Math.log(number) / Math.log(base) ) +1

2’s complement is first invert all the bits and then add 1;

Is number is power of two ..

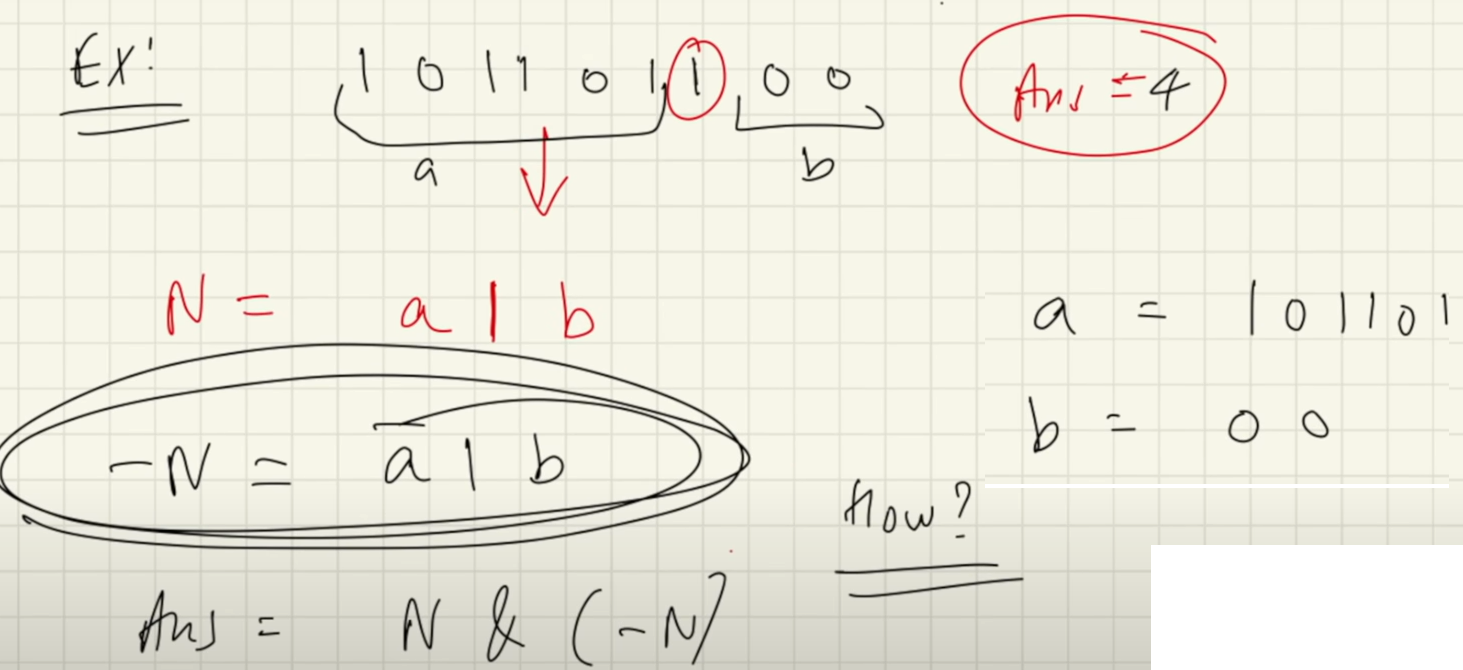
N & (N-1) .. mean just minus one which make this number 1 length smaller and all number will be 1. And if we take & operations then it will be zero then it is power of 2. Else not. (exception n=0 )

1 0 0 0

0 1 1 1

0 0 0 0

Find right most set bit..



-N is known as 2’s complement.

-x = ~x + 1; (it is two’s 2’s complement formula)

Remove Last Set Bit:

n & (n-1) = mean removed the last set bit

00010100       &                (n = 20)  
00010011                        (n-1 = 19)  
~~~~~~~~  
00010000 16

Find the Last Set Bit:

Solution#1 ( n & -n)

00…0010100   &                    (n = 20)  
11…1101100                        (-n = -20)  
~~~~~~~~~~  
00…0000100

Solution#2

00010100    &                (n = 20)  
00010011                     (n-1 = 19)  
~~~~~~~~  
00010000    ^                (XOR result number with n)  
00010100  
~~~~~~~~  
00000100  ——  rightmost set bit will tell us the position

<https://www.techiedelight.com/bit-hacks-part-1-basic/>

Square Root of number using Newton formula.



