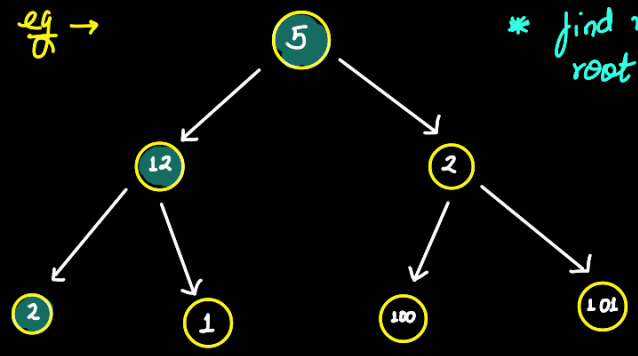
[Greedy Algorithms]:

↳ chooses the best option for that particular moment.

eg →



* find maximum sum from root to leaf node 5 के बाद best desc. 12 then 2

so max sum = 5+12+2 = 19

Which is definitely not correct but that is how greedy technique works.

↳ local optimum

* Generally greedy mein hum sort, max/min, priority queue, set, custom comparator etc. se deal karte hain.

(Q1) ⇒ N-meetings in one room : (activity selection)

Fractional Knapsack :

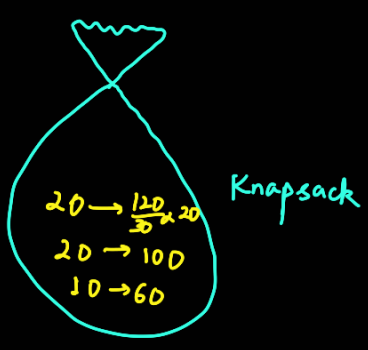
values[] = {60, 100, 120} N = 3

weights[] = {10, 20, 30} W = 50

val/wt. = 6, 5, 4 → sort it.

T.C. = N log N + N
(sorting) ↳ (iteration)

S.C. = O(1)



Ques ⇒ [10, 5, 2, 6] k = 100

i ↑ ↑ j

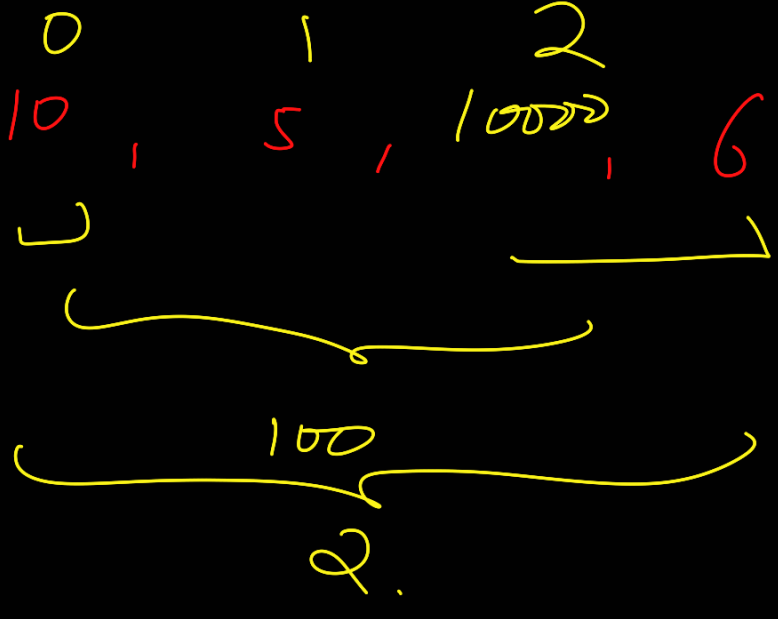
cnt = 0; product = nums[j];

```
while ( j < size) {
    if (product < k) {
        cnt++;
        j++;
    }
    product *= nums[j];
    while (product > k) {
        product /= num[i];
        i++;
    }
}
return cnt;
}
```

cnt = 1 1 1

p = ~~10~~ ~~50~~ ~~100~~

10



If sub. array product < k ;
each no < k in that subc

1

10

10

5

10, 5

1 + 2 + 2 + 3

5, 2

2

= 8

8, 2, 6

2, 6

6

1, 1, 1, 0

R = 1.

product = 1

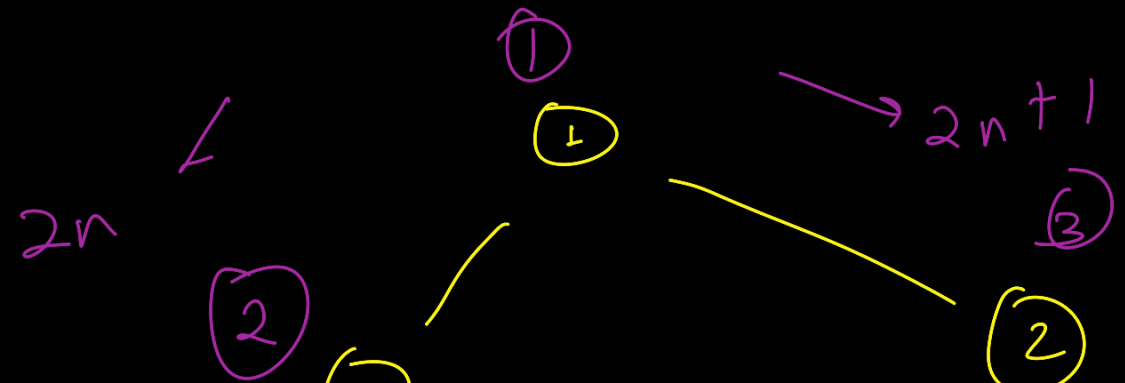
0 - 1 + 1

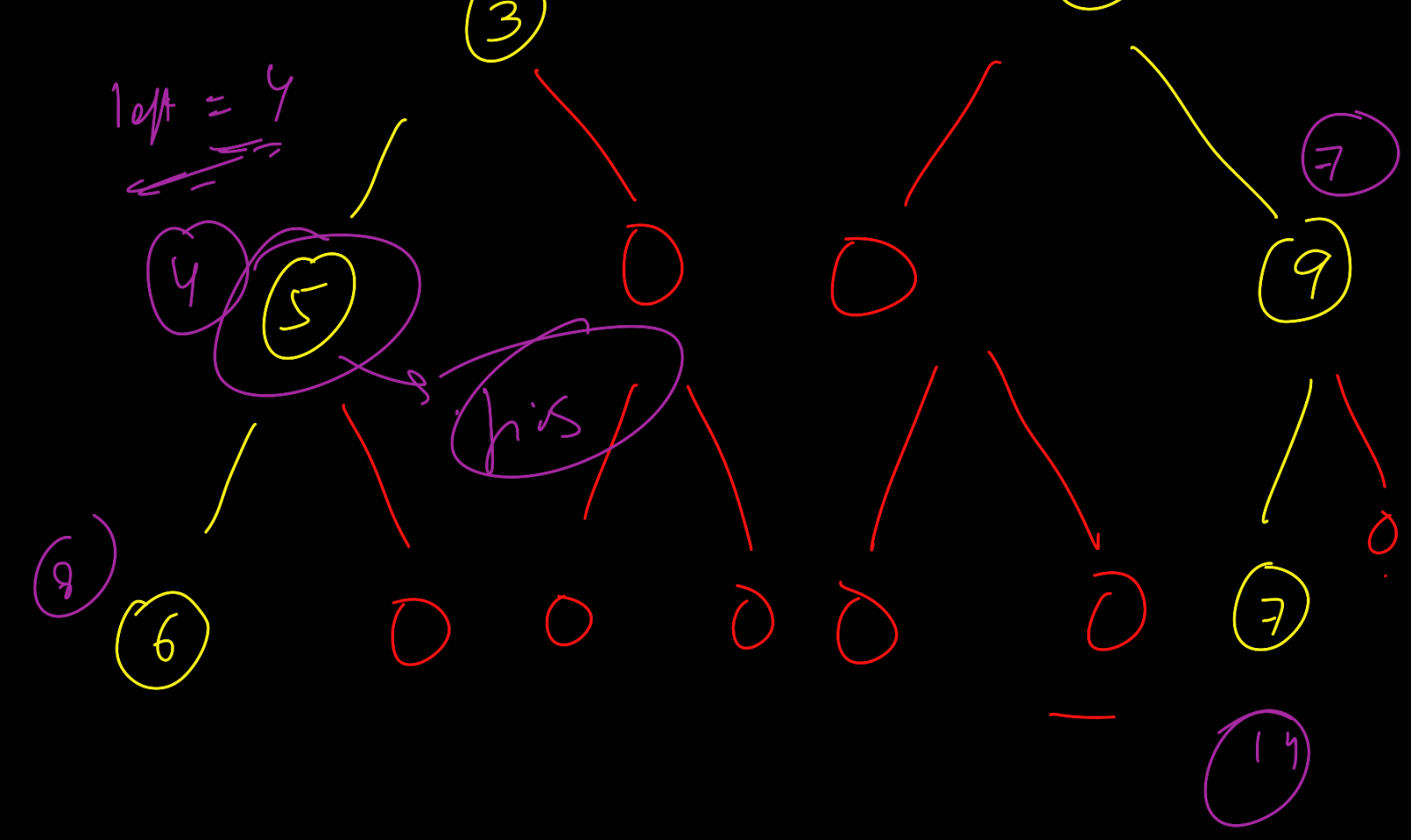
0

vertical orde

each level pe

leftmost node - Right most node





14 - 8 + 1