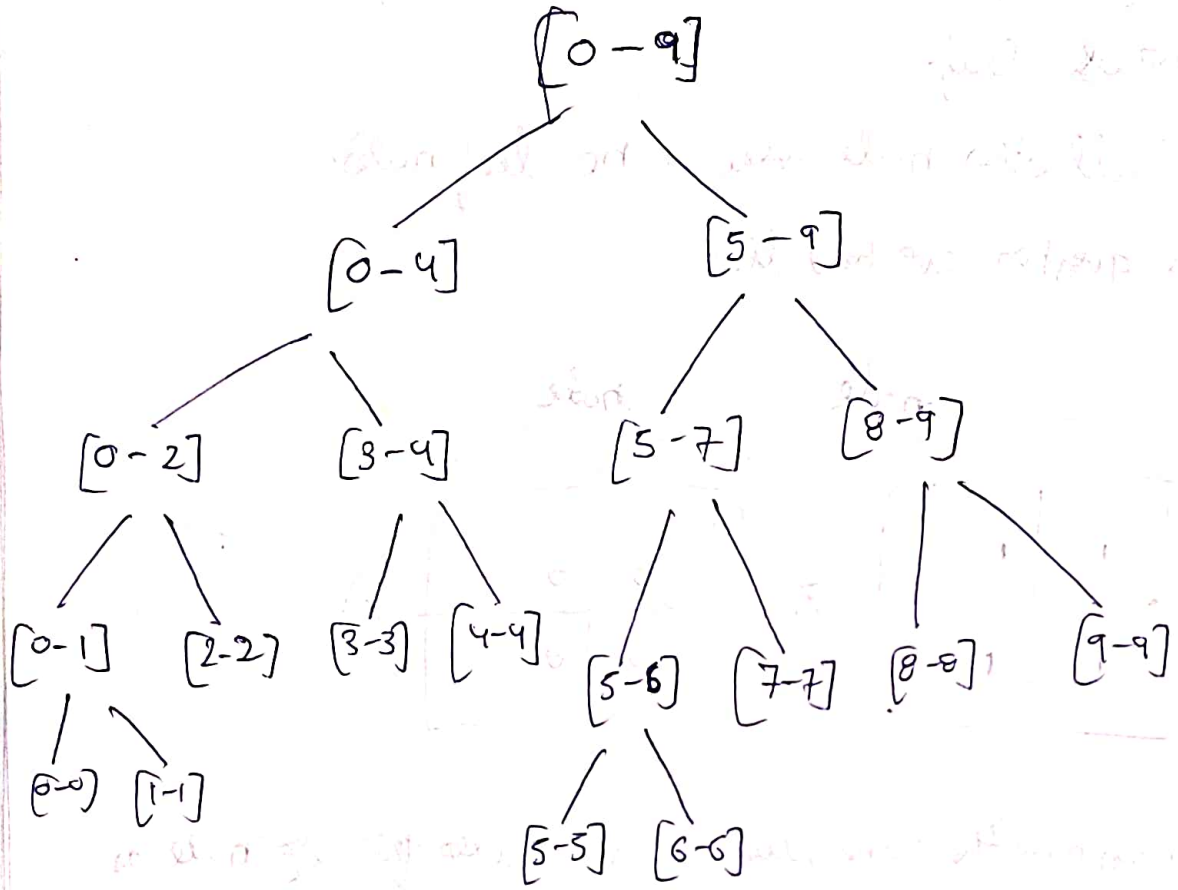
Normal segment tree representation

[0-9]

Dynamic segment tree Learning



Dynamic segment tree Node

node
l = 0
r = 9
val = 0.

Update comes

We want to solve the question of max node

(with pointed updates)

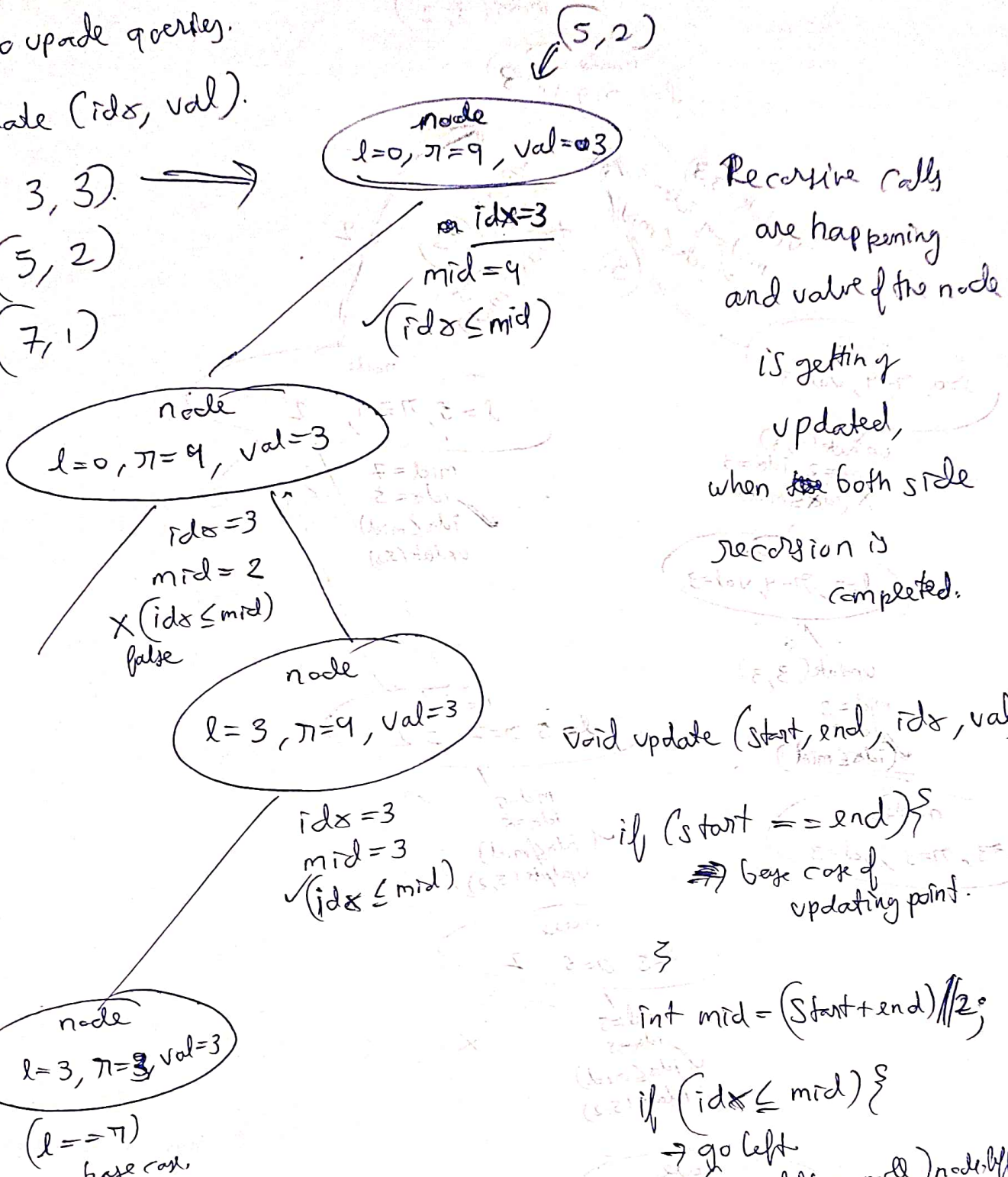
two update queries.

update (idx, val).

(3, 3)

(5, 2)

(7, 1)



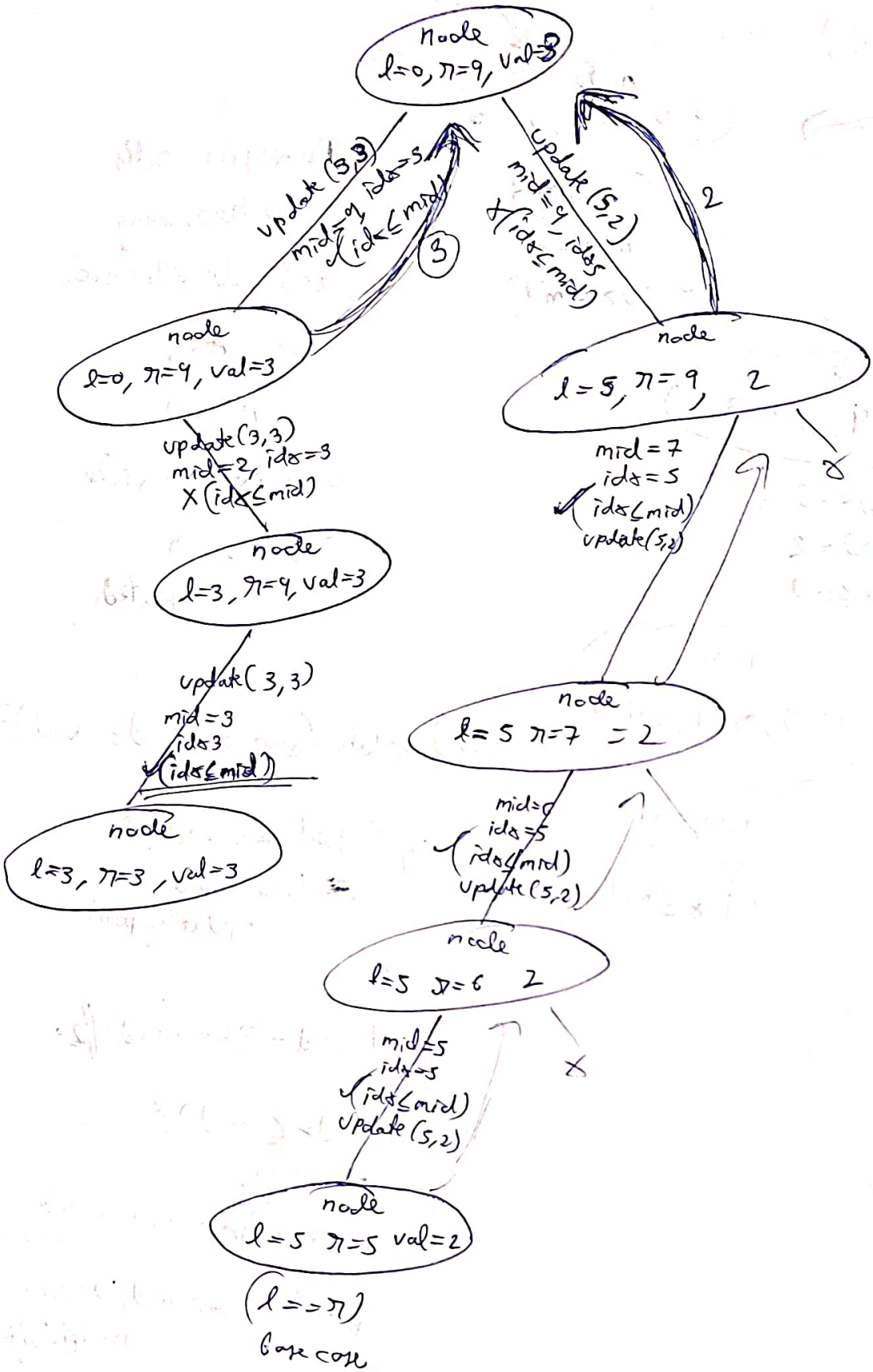
Recursive calls are happening and value of the node is getting updated, when both side recursion is completed.

```
void update (start, end, idx, val){
  if (start == end){
      => base case of updating point.
  }
  int mid = (start+end)//2;
  if (idx ≤ mid){
      → go left
      if (node.left == null) node.left = new node();
      update (start, mid, idx, val, node.left)
  }else{ → go right
      if (node.right == null) node.right = new node();
      update (mid+1, end, ...
  }
```

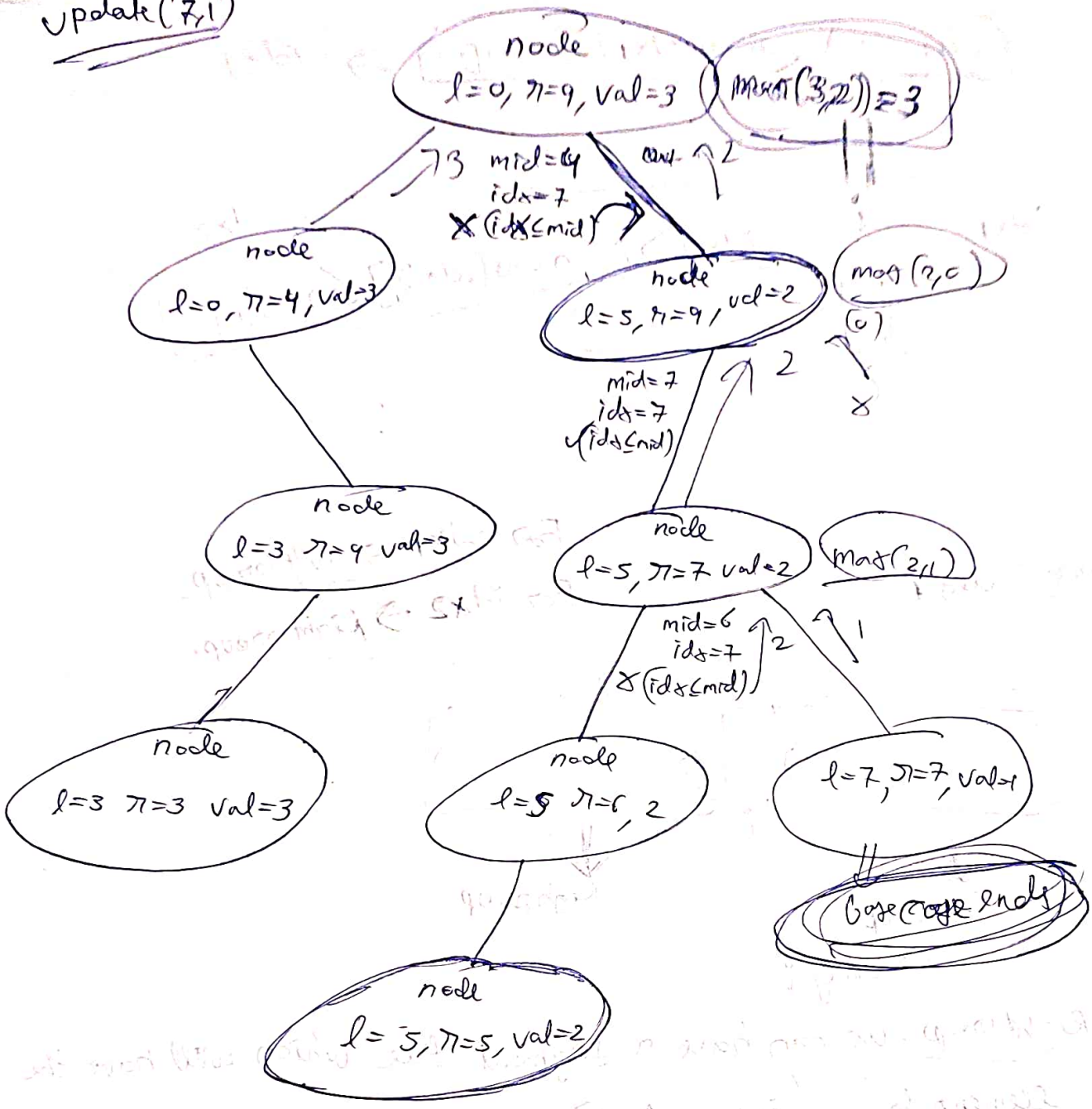
merging is happening

```
node.val = getVal(node.left) + getVal(node.right);
```

max(3, 2)


node
l=0, r=9, val=3
update(3,3)
mid=4, ids=3
X(ids<=mid)
3
update(5,2)
mid=4, ids=5
X(ids<=mid)
2
node
l=0, r=4, val=3
node
l=5, r=9, 2
update(3,3)
mid=2, ids=3
X(ids<=mid)
mid=7
ids=5
(ids<=mid)
update(5,2)
node
l=3, r=4, val=3
node
l=5 r=7 =2
update(3,3)
mid=3
ids3
X(ids<=mid)
mid=6
ids=5
(ids<=mid)
update(5,2)
node
l=3, r=3, val=3
node
l=5 r=6 2
mid=5
ids=5
(ids<=mid)
update(5,2)
node
l=5 r=5 val=2
(l==r)
Base case

update(7,1)



New comes the query part

Base Cases are 3 → ① if node is null ⇒ return default value

② if node is out of range (r < start || l > end)
⇒ return defaultValue

③ if node is completely within range
if (l ≥ start && end ≥ r)
→ return this node's value

Else → go to left and right and merge the values.