Tree Intuition

Strukdat RKA (N)

Abstraction

Tree

↳ Root , Leaf

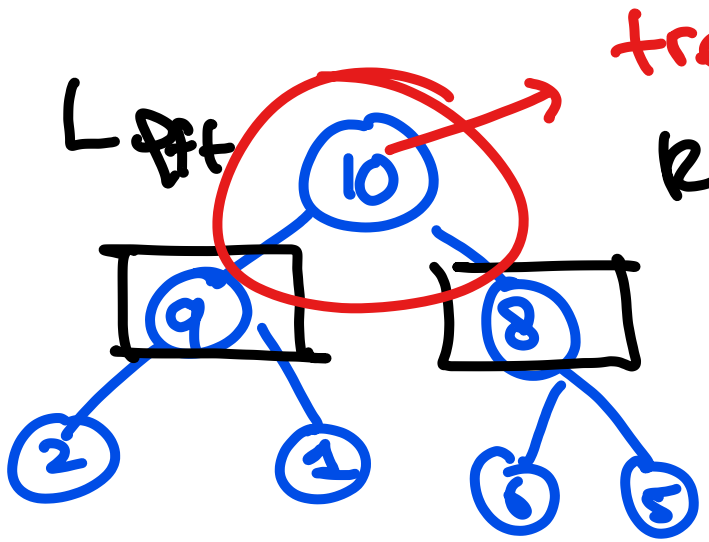
Parent , children

Binary ← Left

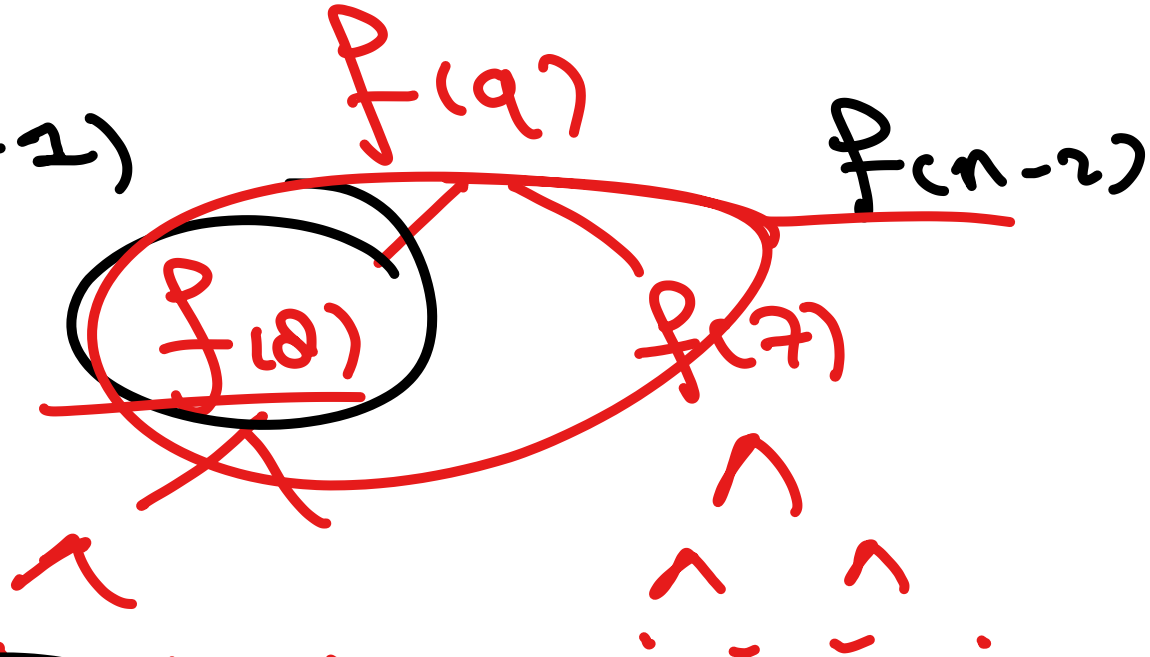
Right

Sum all of nodes value in Binary Tree

$$f(n) = f(n-1) + f(n-2)$$



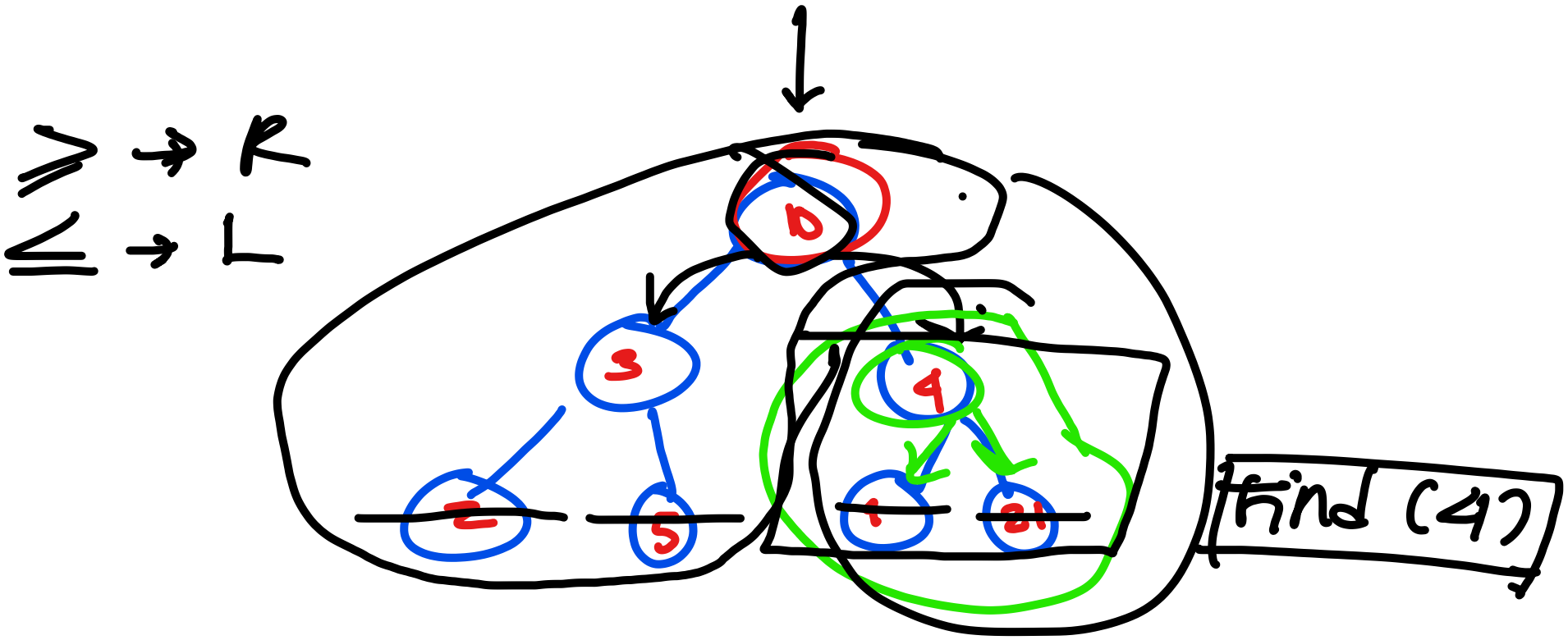
f(n-1)



sum(tree) =

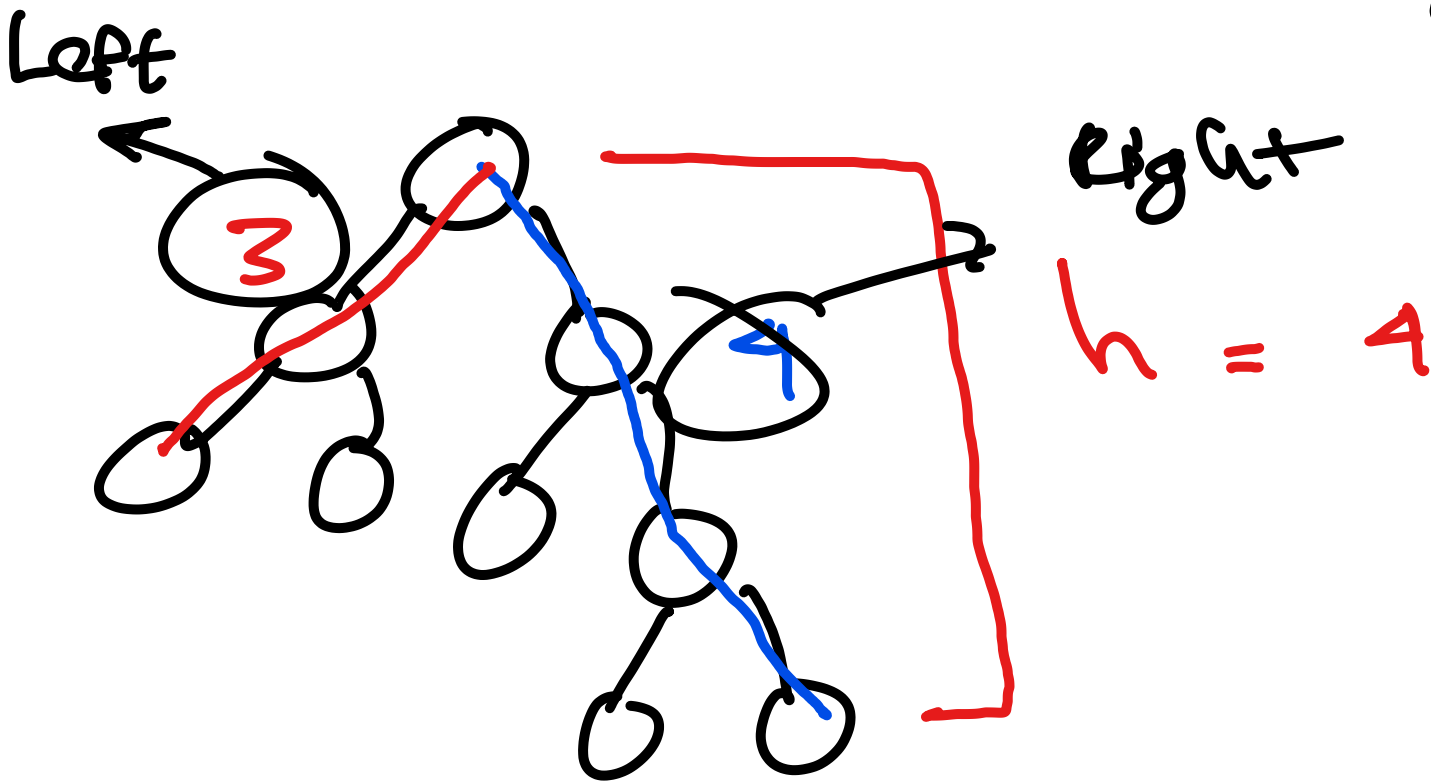
sum(tree["left"]) +
sum(tree["right"]) +
tree["val"]

Max (tree) = Max(Max (tree["left"]),
Max (tree["right"]))

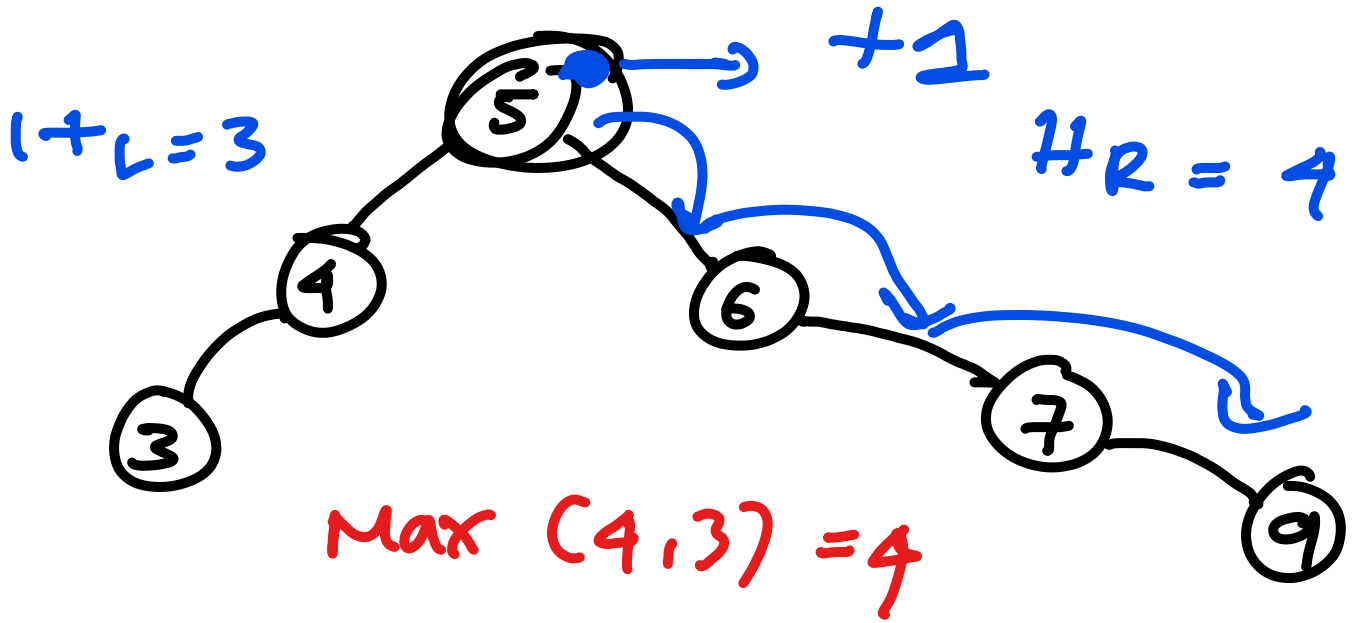


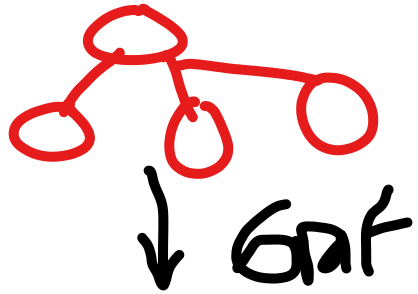
dfs (node) :

Visit connection ⇒ visit left
visit right



5 4 6 7 9 3



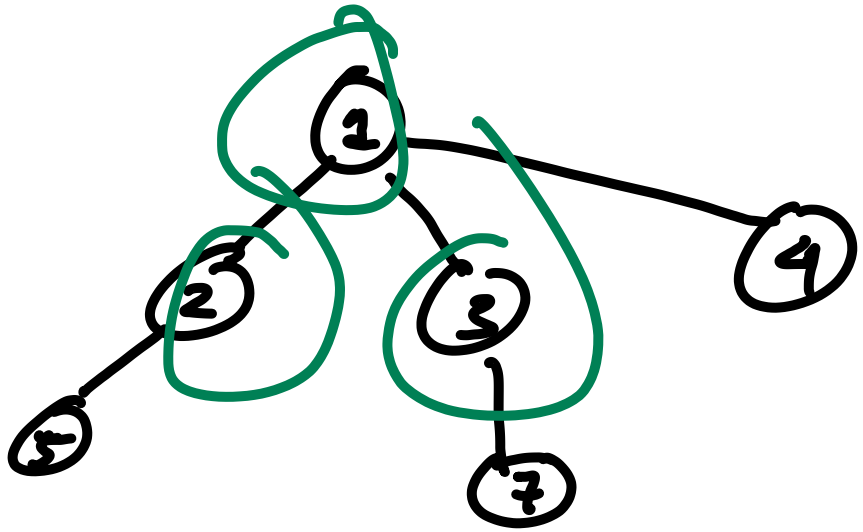
N - Any → 3 - Any → 

4 - Any →

yg punya anak

yg jadi anak

1 2
1 3
1 4
2 5
3 7



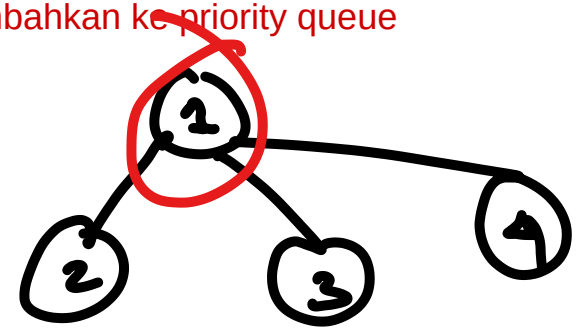
key ← { 1:[2,3,4] } array / list

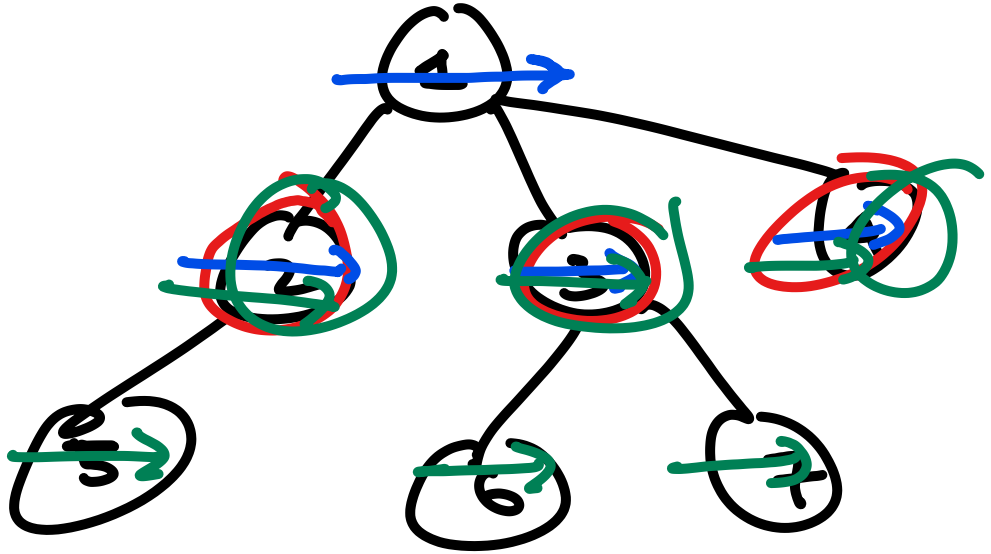
value

```
def bfs(adj, start):
    visited = set()
    priority_queue = [(0, start)]  # (weight, node)

    while priority_queue:
        weight, node = heapq.heappop(priority_queue)  # Ambil node dengan bobot terkecil
        if node not in visited:
            print(node, end=' ')
            visited.add(node)

            for neighbor_dict in adj[node]:  # Akses dictionary dalam list
                for neighbor, w in neighbor_dict.items():  # Ambil tetangga dan bobotnya
                    if neighbor not in visited:
                        heapq.heappush(priority_queue, (w, neighbor)) # Tambahkan ke priority queue
```

Mark

neighbor.key * ...





BFS

pq heap q

1 - 3 - 2 - 4 -

5 - 6 - 7