CS & IT ENGINEERING

Algorithm

[redacted]

Heap Algorithms

Lecture No.- 02

By- Dr. Khaleel Khan Sir

Topics to be Covered

Topic

Sets

Representation of Sets

Set Operations : Union and Find

Recursion Tree Method for Solving D and C Recurrences

(i) $T(n) = a \cdot T(n/b) + f(n)$

- Back Subst.
- Master Method
- Recursion Tree

(ii) $T(n) = T(\alpha n) + T((1-\alpha)n) + f(n) \rightarrow$ Recursion Tree

$T(n) = T(n/3) + T(2n/3) + n$

(iii) $T(n) = T(\alpha n) + T(\beta n) + T(\gamma n) + f(n)$

$\rightarrow$ Recursion Tree

Ex: $T(n) = T(n/2) + T(n/3) + T(n/4) + n$

1) Back Subst. — Value
— order

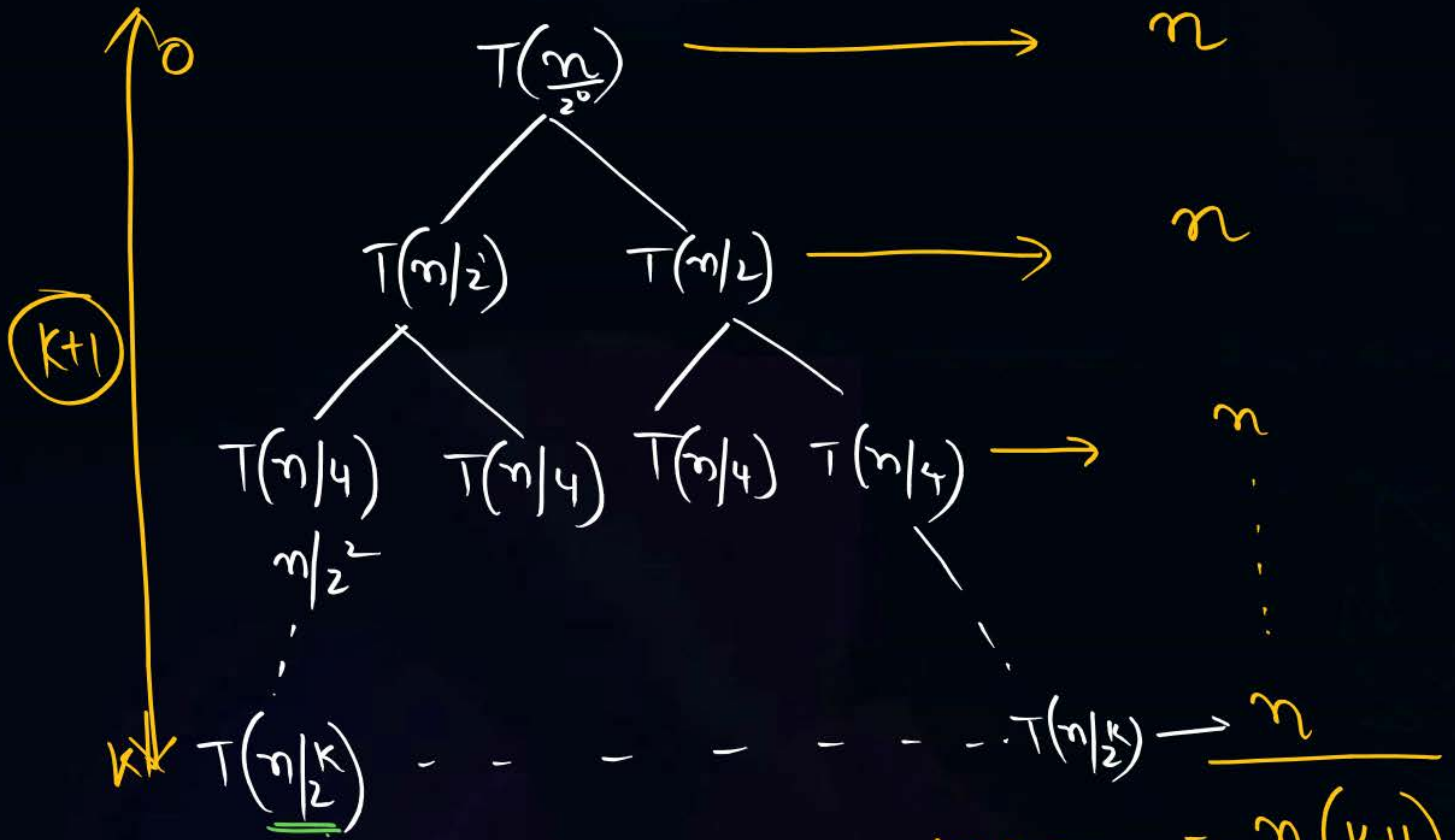
$$T(n) = T(n-1) + \lfloor \sqrt{n} \rfloor$$

2) Master Method → order

3) Recursion Tree → order

1) $T(n) = 2 \cdot T(n/2) + n$ cost $\qquad T(1) = 1$

Cost



$O(n \cdot \log n)$

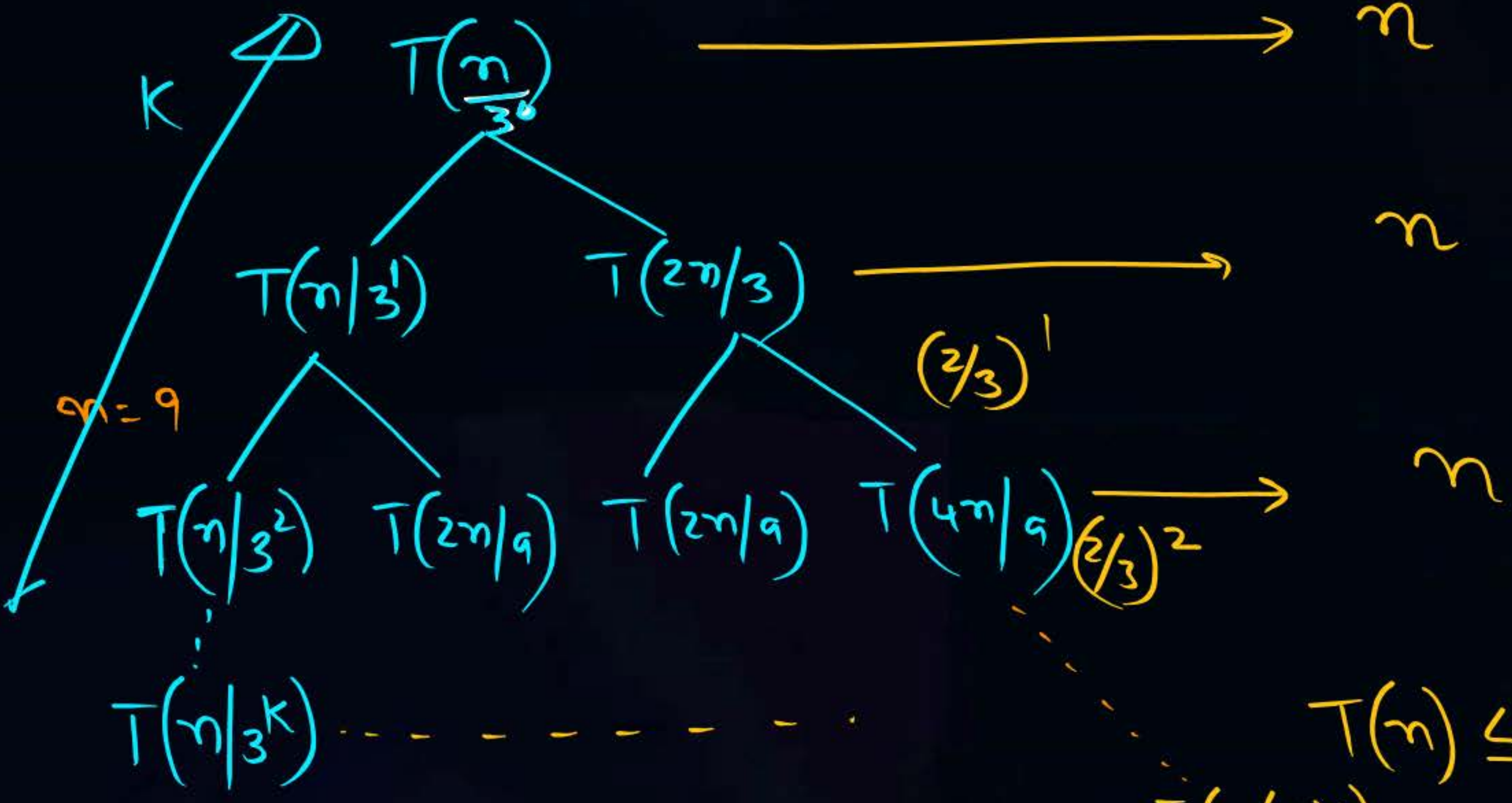
$$\boxed{\frac{n}{2^K} = 1}$$

$n = 2^K$

$K = \log_2 n$

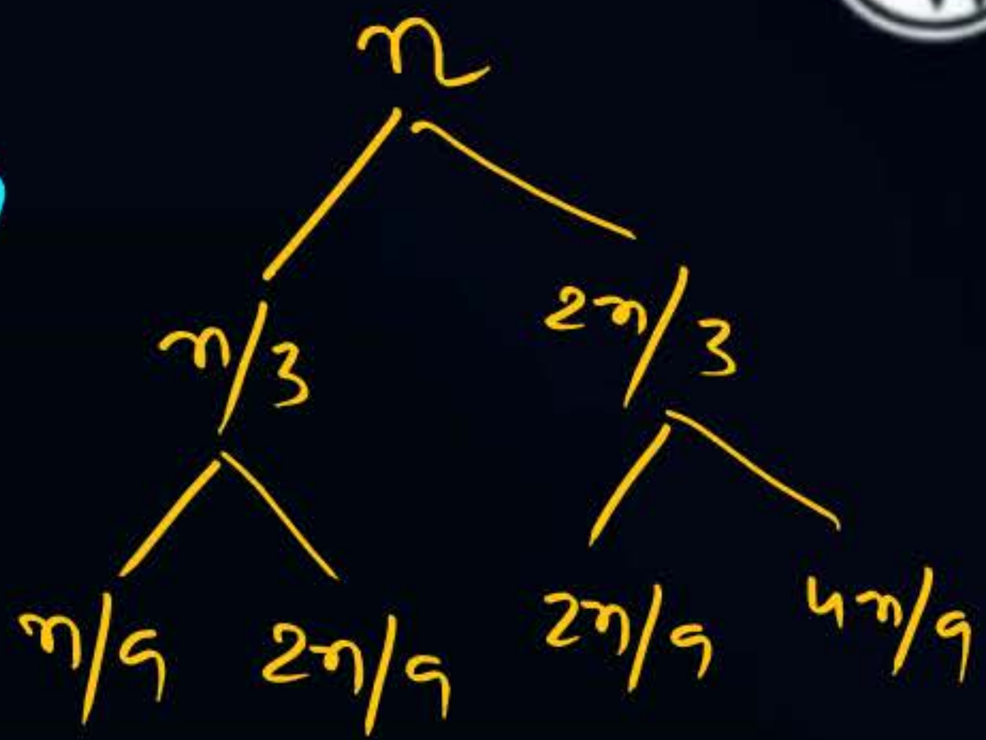
Total Cost (Time) $= n(K+1) = n(\log n + 1)$

2) $T(n) = T(n/3) + T(2n/3) + n$



$\frac{n}{3^K} = 1$

$\Rightarrow K = \log_3 n$



$n \cdot (2/3)^K = 1$

$K = \log_{3/2} n$

$T(n) \le n * K$

$\le n * \log_{3/2} n$

$T(n) \ge n * K$

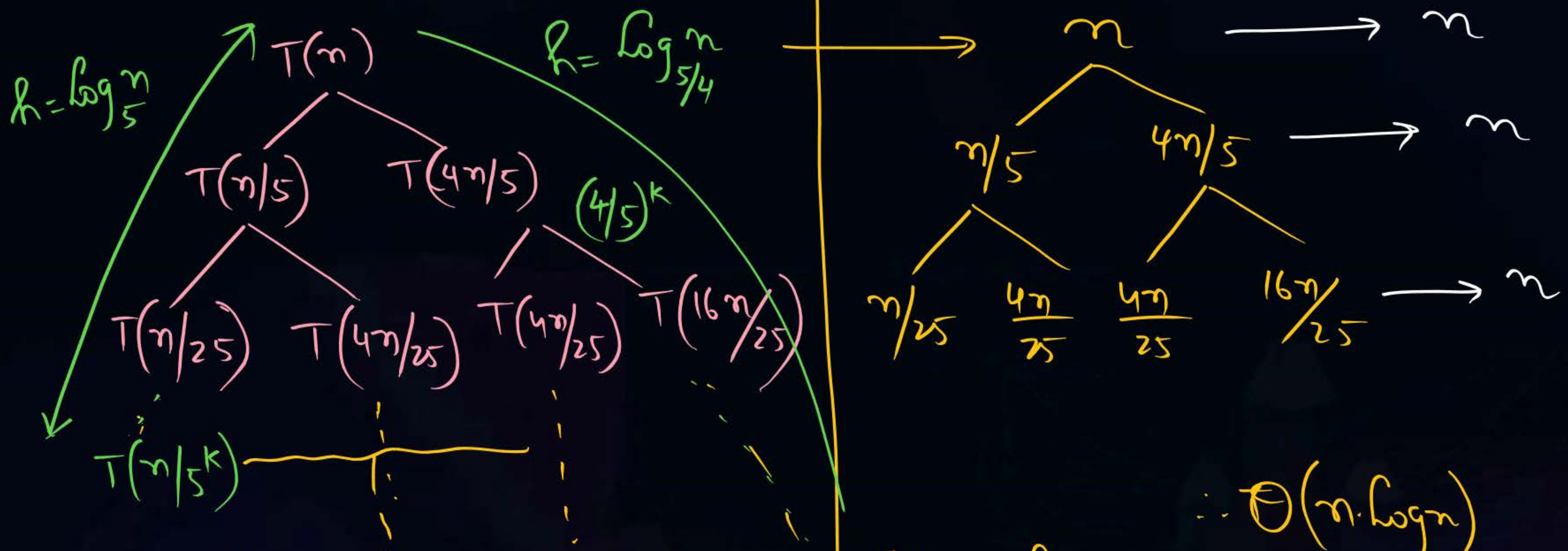
$\ge n \cdot \log_3 n$

$\Omega(n \cdot \log_3 n)$

$\boxed{T(n) = \Theta(n \log n)}$

$T(n) = O(n * \log_{3/2} n)$

3) $T(n) = T(n/5) + T(4n/5) + n$



$\therefore \Theta(n \cdot \log n)$

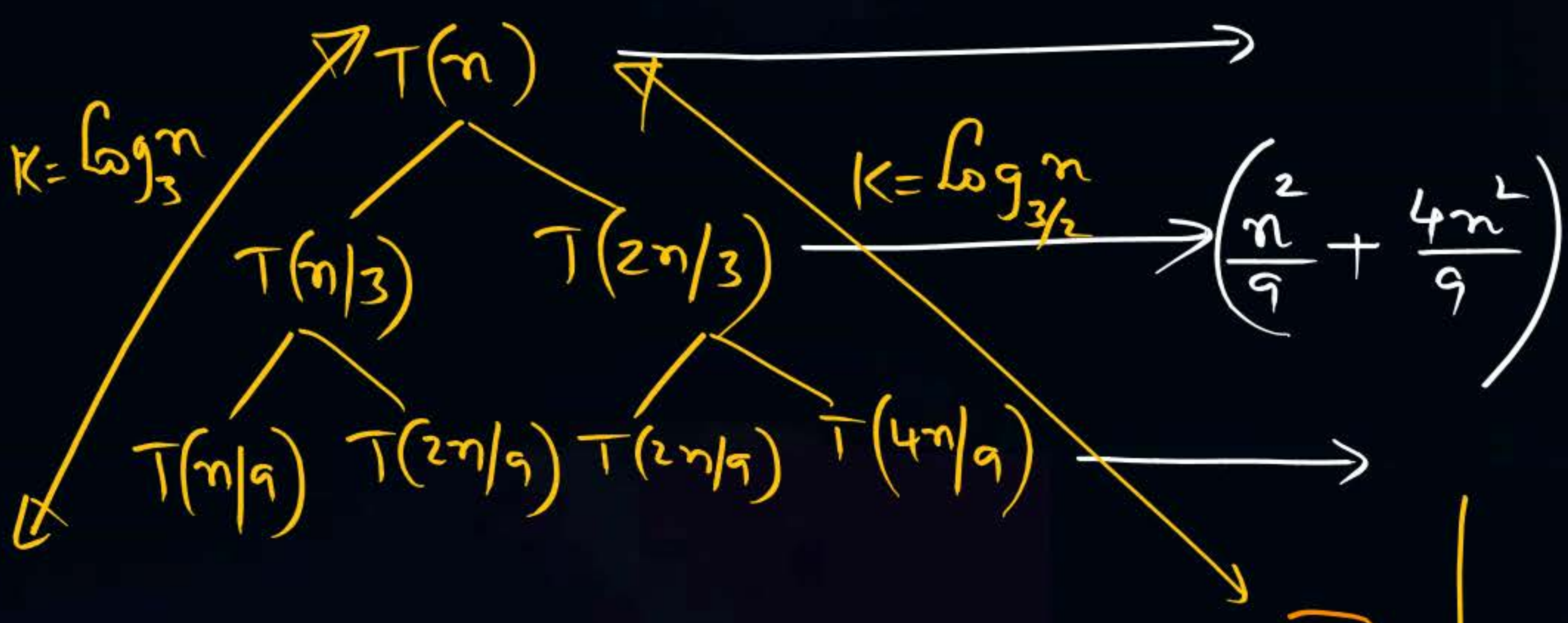
$T(n) \geq n * \log_5 n$

$\Omega(n \log n)$

$T(n) \leq n \cdot \log_{5/4} n$

$O(n \cdot \log n)$

4) $T(n) = T(n/3) + T(2n/3) + n^2 \implies T(n) = \Theta(n^2)$ ✓



$\left(\frac{5}{9}\right)^0 n^2$

$\left(\frac{5}{9}\right)^1 n^2$

$\left(\frac{5}{9}\right)^2 n^2$

$S_n = \frac{a(1-r^n)}{1-r}$

$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}$ $\quad x < 1$

$K = \log_{3/2} n$

$$L.S.T = \left[n^2 + \frac{5}{9}n^2 + \left(\frac{5}{9}\right)^2 n^2 + \cdots + \left(\frac{5}{9}\right)^K \cdot n^2\right]$$

$$= n^2 \cdot \sum_{i=0}^{K} (5/9)^i \sim n^2 \cdot \left(\frac{1}{1-\frac{5}{9}}\right)$$

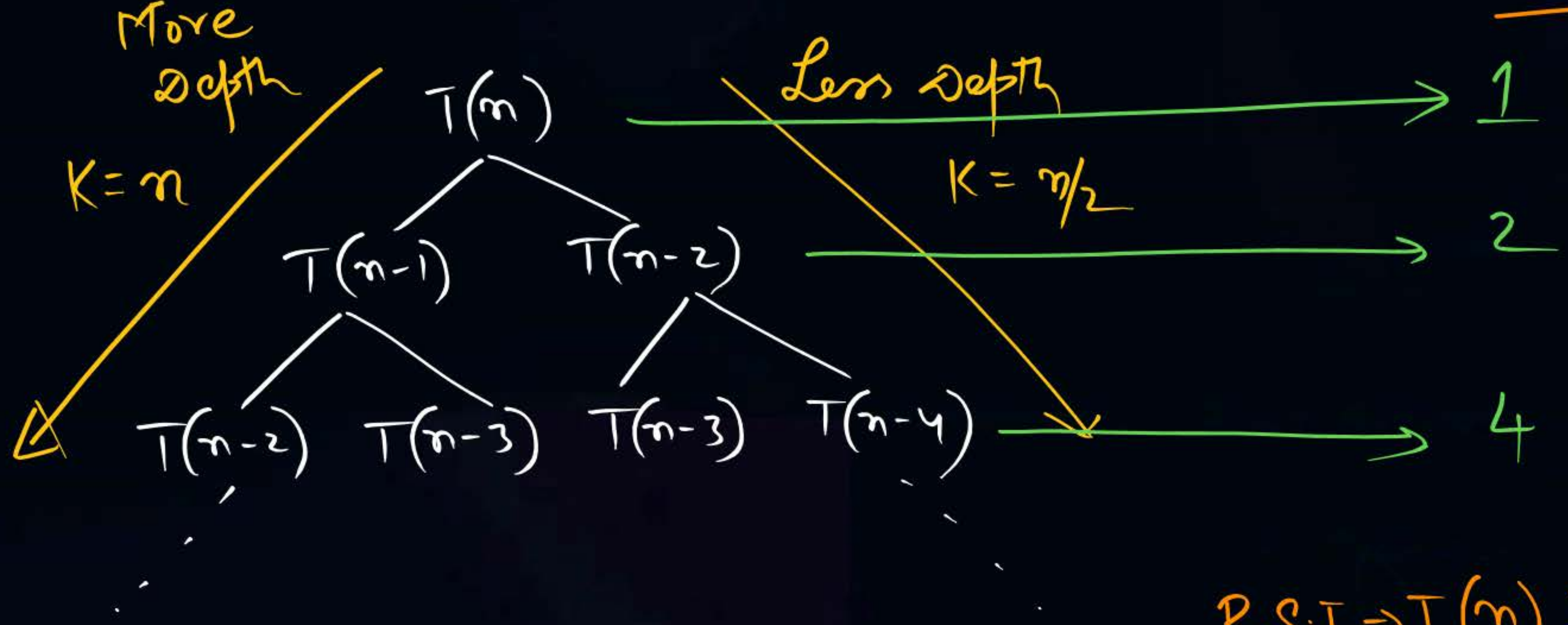
$$T(n) \geq n^2 \cdot c = \Omega(n^2)$$

$$R.S.T = n^2 + \left(\frac{5}{9}\right) n^2 + \left(\frac{5}{9}\right)^2 n^2 + \cdots + \left(\frac{5}{9}\right)^K \cdot n$$

$$= n^2 \cdot \sum_{i=0}^{K} (5/9)^i$$

$$= n^2 \cdot c_2 = O(n^2)$$

5) $T(n) = T(n-1) + T(n-2) + 1$ Is it that $T(n) = \Theta(2^n)$?



$$S_n = \frac{a(r^n - 1)}{r-1}$$

$$= \frac{1(2^{n+1} - 1)}{2-1}$$

$$= 2^{n+1}$$

$$L.S.T = \sum_{i=0}^{n} 2^i = \left(2^{n+1} - 1\right)$$

$$L.S.T \left(T(n) \le 2^{n+1} - 1 \right.$$

$$O(2^n)$$

$$R.S.T \Rightarrow T(n) \ge \sum_{i=0}^{n/2} 2^i = \frac{1(2^{k+1} - 1)}{2-1}$$

$$= 2^{n/2 + 1} - 1$$

$$T(n) = \Omega\left(2^{n/2}\right)$$

$$2^n \quad vs \quad 2^{n/2}$$

$$\cancel{2^{n/2}} \times 2^{n/2} \quad vs \quad \cancel{2^{n/2}}$$

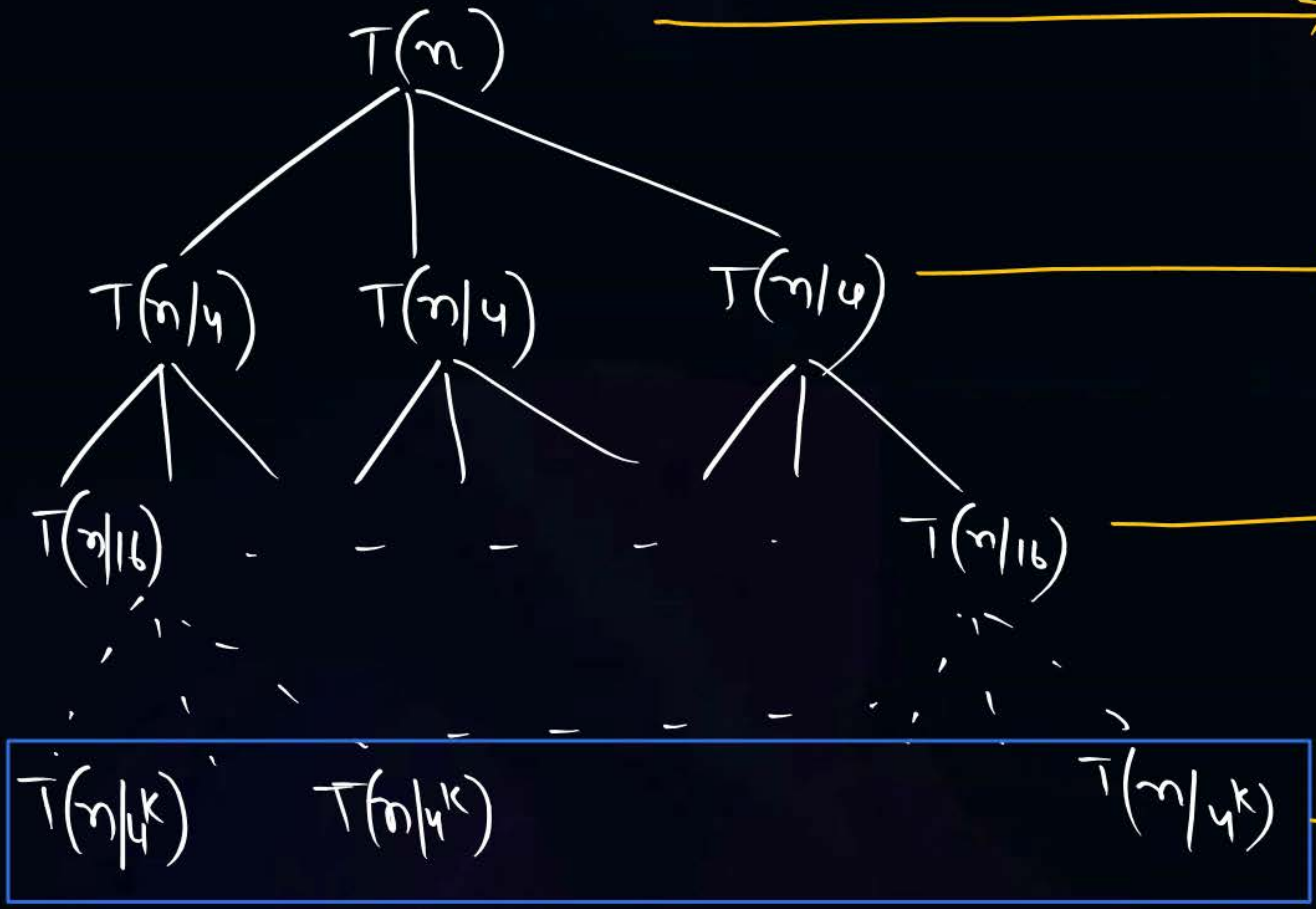
$$2^{n/2} \quad vs \quad 1$$

$$>$$

$$T(n) = 3\cdot T(n/4) + n^2$$

$$\frac{n}{4^k} = 1$$

$$\Rightarrow k = \log_4 n$$



$$n^2$$

$$\left(\frac{3}{16}\right) n^2$$

$$\left(3/16\right)^2 n^2$$

$$\vdots$$

$$\left(3/16\right)^{k-1} \cdot n^2$$

$$3^k \cdot T(1)$$

Time :

$$T(n) = \left[n^2 + \left(\frac{3}{16}\right) n^2 + \left(\frac{3}{16}\right)^2 n^2 + \cdots + \left(\frac{3}{16}\right)^{K-1} \cdot n^2\right] + 3^K \cdot T(1)$$

$$= n^2 \cdot \sum_{i=0}^{K-1} \left(\frac{3}{16}\right)^i + 3^{\log_4 n}$$

$$T(n) = c \cdot n^2 + n^{\log_4 3}$$

$$= O(n^2) ; \Omega(n^2)$$

$$\therefore T(n) = \Theta(n^2) \checkmark$$

H/w Q1) $T(n) = T(n/2) + T(n/3) + T(n/4) + n$

MCQ

H/w 2

Consider the following recurrence relation:

$$T(n) = \begin{cases} T\left(\frac{n}{2}\right) + T\left(\frac{2n}{5}\right) + 7n & \text{if } n > 0 \\ 1 & \text{if } n = 0 \end{cases}$$

Which one of the following options is correct? [GATE-2021: 2M]

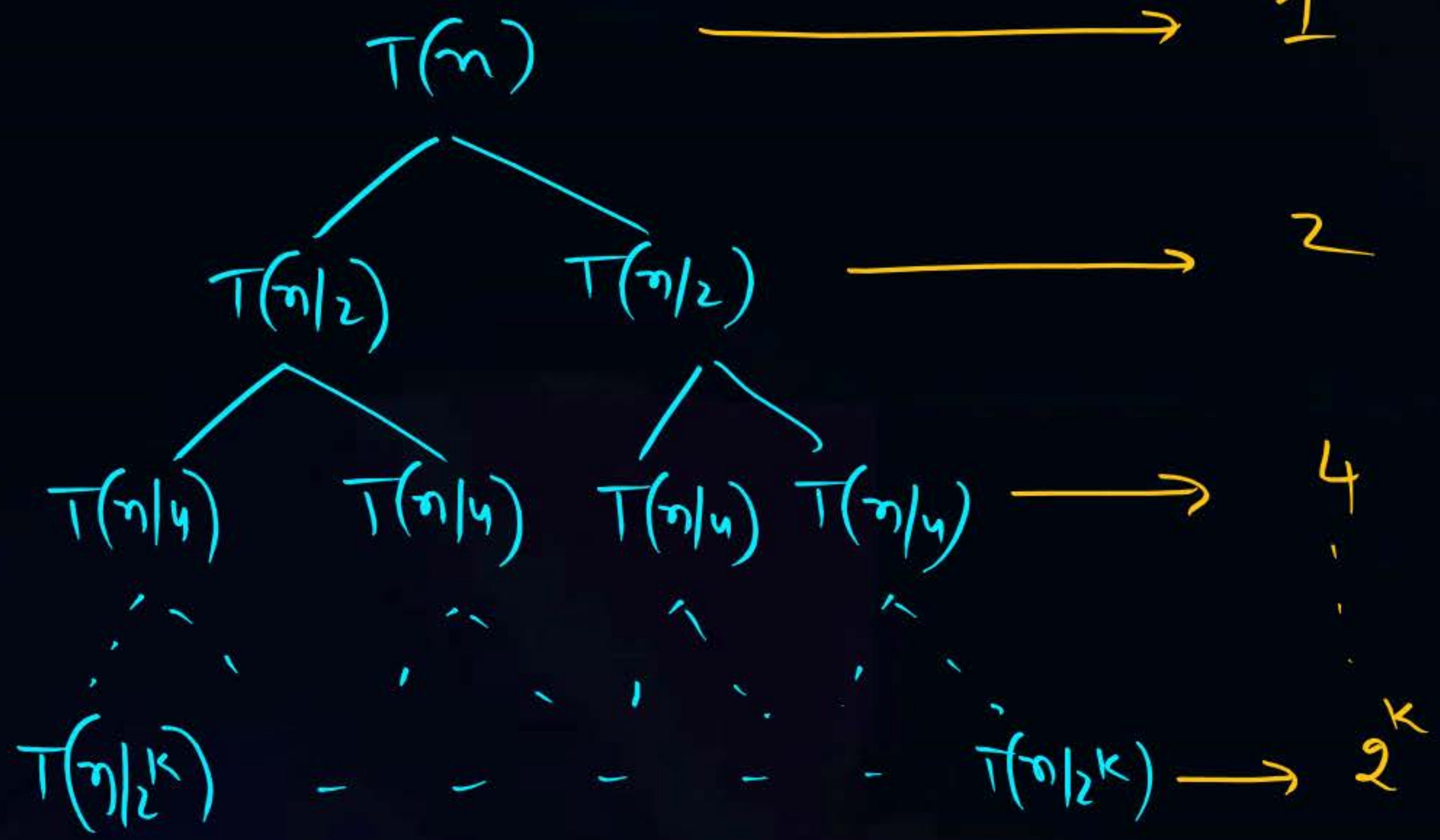
(A) $T(n) = \theta(n\log n)$

(B) $T(n) = \theta(n^{5/2})$

(C) $T(n) = \theta(n)$

(D) $T(n) = \theta\left((\log n)^{5/2}\right)$

$$T(n) = 2 \cdot T(n/2) + 1$$



$$\sum_{i=0}^{K+1} 2^i$$

$$\frac{n}{2^K} = 1$$

$$K = \log n.$$

$$\Rightarrow \left(2^{\log n + 1} - 1\right)$$

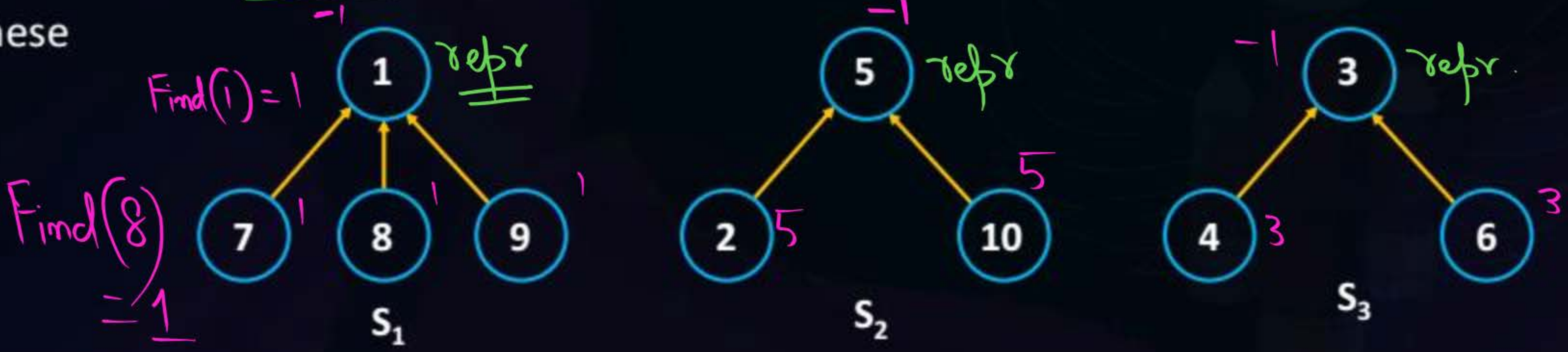
$$O(n)$$ ✓

Topic : Sets

$S = \{1, 2, 5, 8\} \rightarrow$ Indices

It is assumed that the elements of the sets are the numbers 1, 2, 3, ..., n. These number might, in practice, be indices into a symbol table in which the names of the elements are stored. We assume that the sets being represented are pairwise disjoint (that is, if S_i, and S_j, $i \neq j$, are two sets, then there is no element that is in both S_i and S_j).

For example, when n = 10, the elements can be partitioned into three disjoint sets, $S_1 = \{1, 7, 8, 9\}$, $S_2 = \{2, 5, 10\}$, and $S_3 = \{3, 4, 6\}$. Figure shows one possible representation for these



Possible tree representation of sets

Slide 4

Topic : Sets

The operations we wish to perform on these sets are:

1. **Disjoint set union.** If S_i and S_j are two disjoint sets, then their union $S_i \cup S_j$ = all elements x such that x is in S_i or S_j. Thus, $S_1 \cup S_2 = \{1, 7, 8, 9, 2, 5, 10\}$. Since we have assumed that all sets are disjoint, we can assume that following the union of S_i and S_j, the sets S_i and S_j do not exist independently; that is, they are replaced by $S_i \cup S_j$ in the collection of sets.

2. **Find** (i). Given the element i, find the set containing i. Thus, 4 is in set S_3, and 9 is in set S_1.

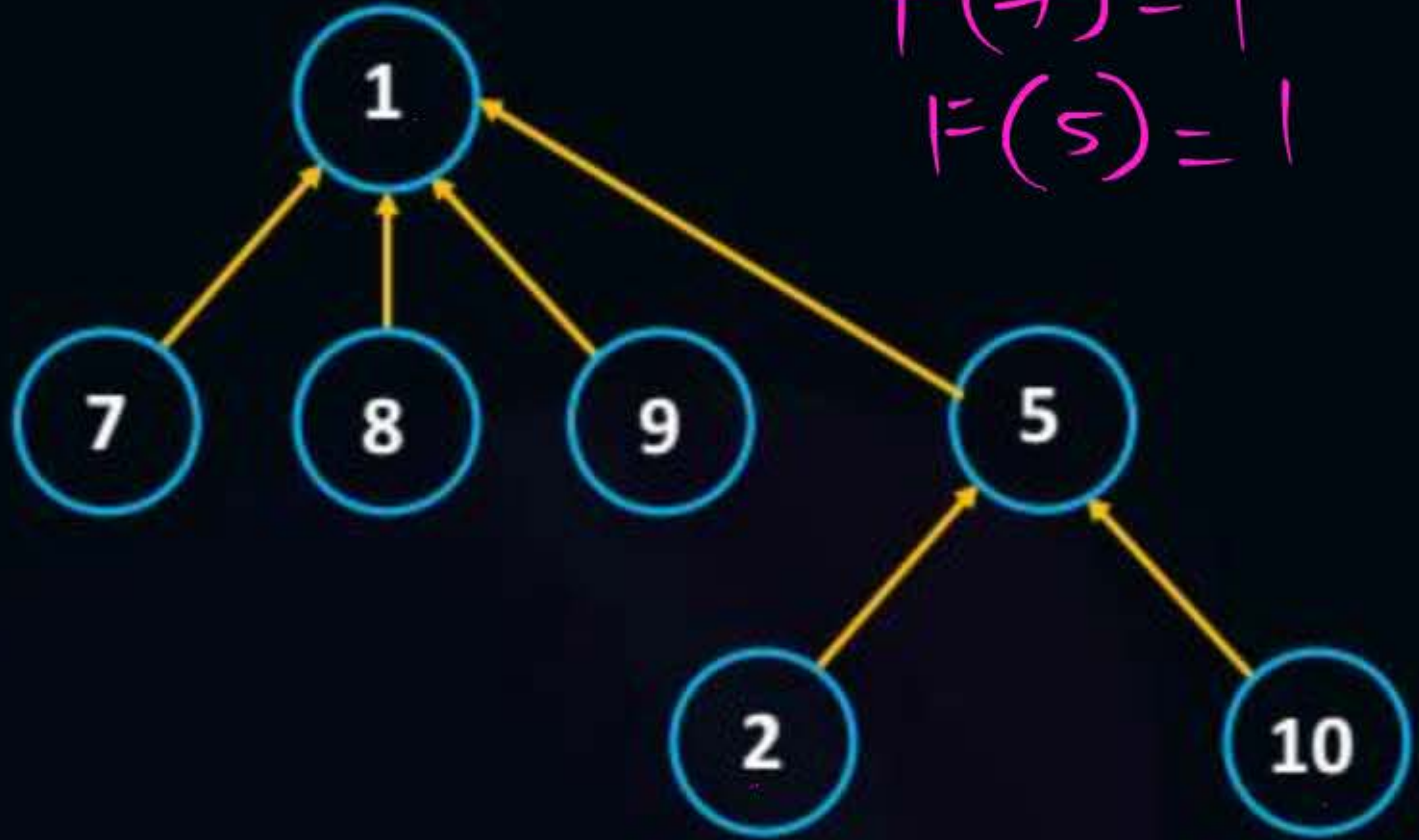
(repr. of the set)

Topic : Sets

$S_1 = \{1, 7, 8, 9, 5, 2, 10\}$

$F(7) = 1$

$F(5) = 1$



$F(2)$

$F(10) = 5$

$F(8) = 5$

$S_1 \cup S_2$ or $S_1 \cup S_2$

Possible representations of $S_1 \cup S_2$

Representation of Sets :

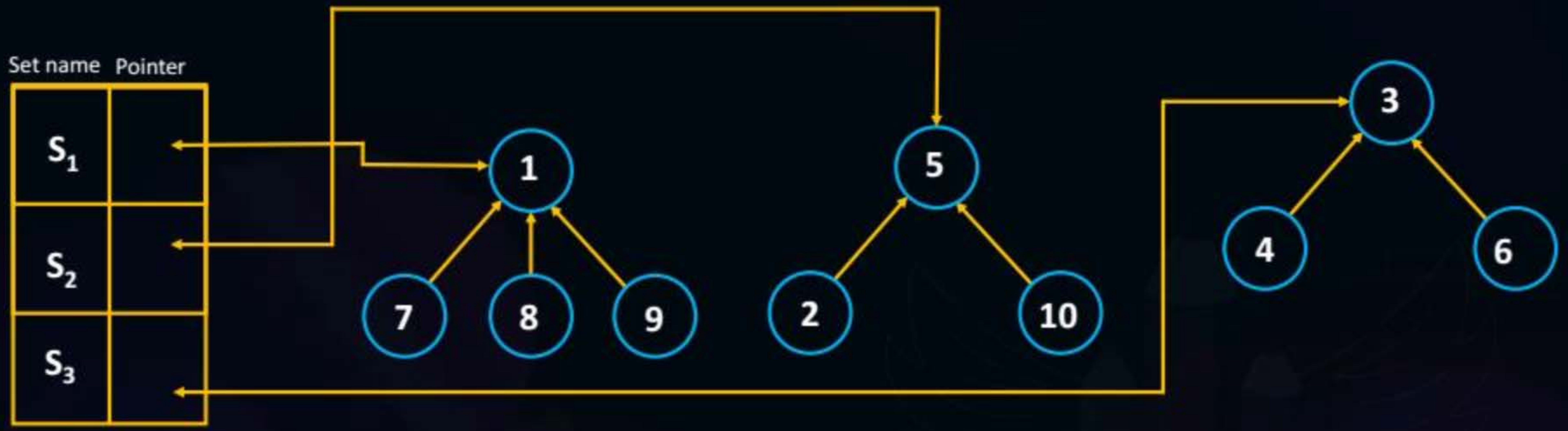
$n = 10$

⇒ Array based Representation :

$S_1 = \{1, 7, 8, 9\}$; $S_2 = \{2, 5, 10\}$ $S_3 = \{3, 4, 6\}$

i	1	2	3	4	5	6	7	8	9	10
P	-1	5	-1	3	-1	3	1	1	1	5

Parent

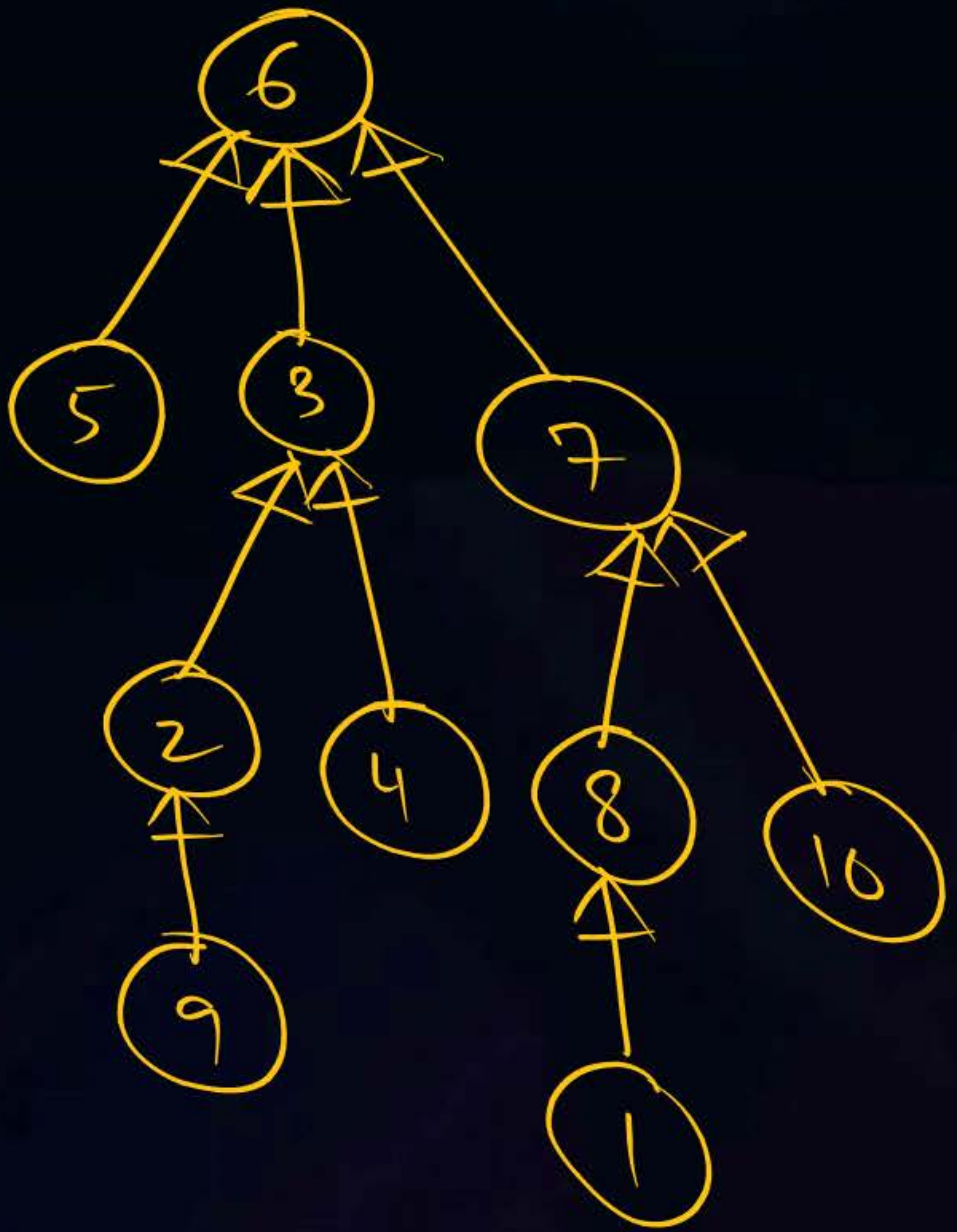


Data representation for S_1, S_2, and S_3

i	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
P	–1	5	–1	3	–1	3	1	1	1	5

Array representation of S_1, S_2, and S_3 of Figure

$S_6 = \{1 \ldots 10\}$



i	1	2	3	4	5	6	7	8	9	10
p	8	3	6	3	6	-1	6	7	2	7

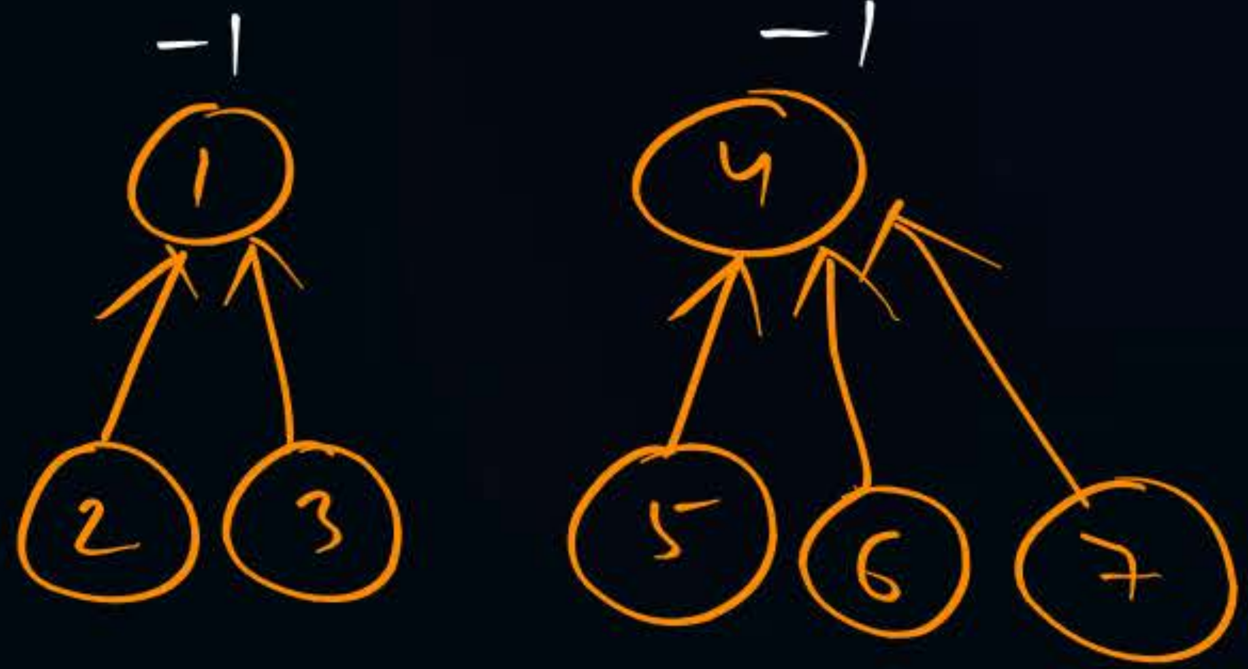
Topic : Sets

```
Algorithm Union (i, j)
{
   p[i]: = j;
}
```

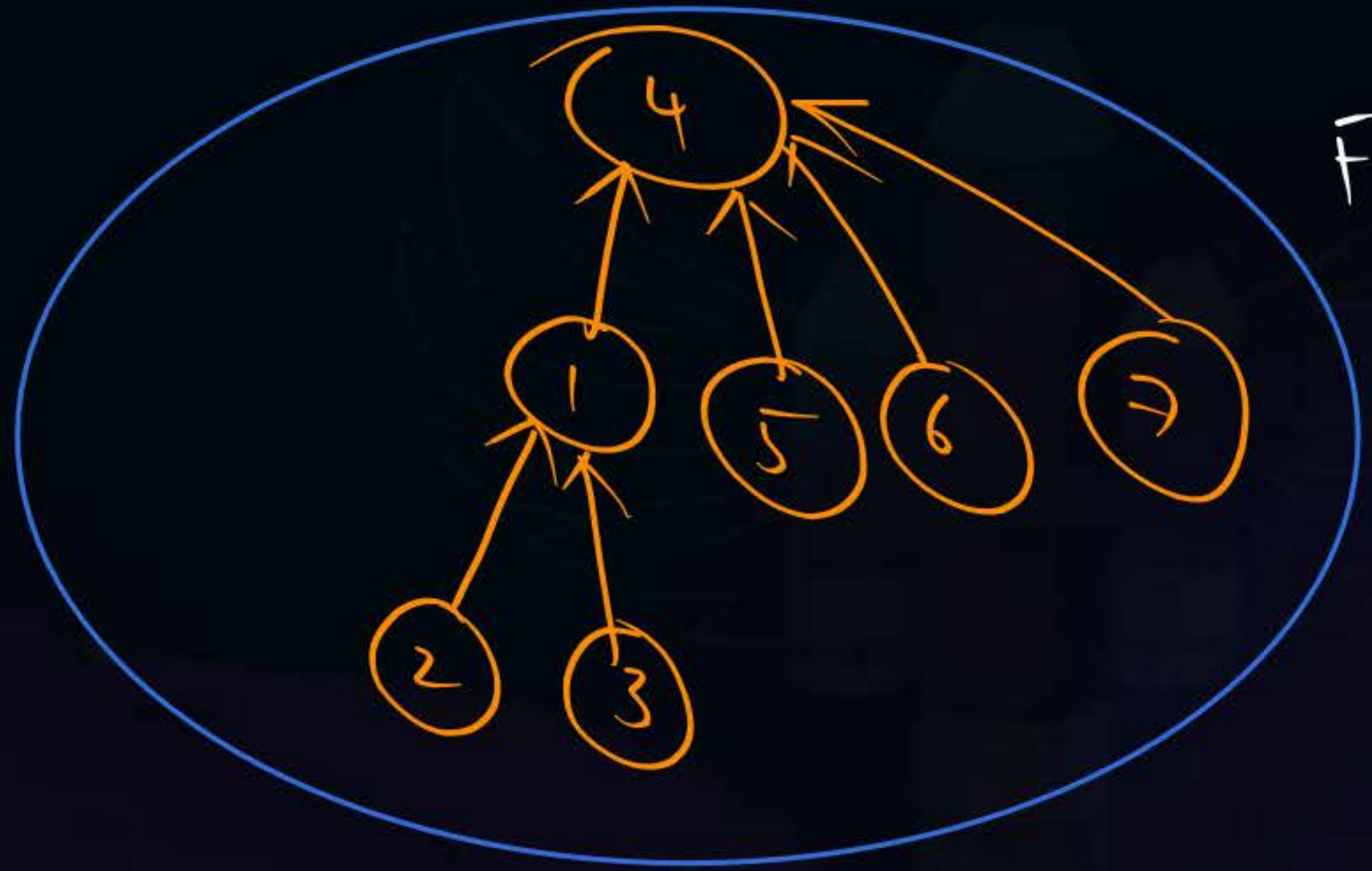
O(1)

```
Algorithm Find(i)
{
        while (p[i] ≥ 0) do i : = p [i];
        return i;
}
```

O(1)



UNION(1, 4)



Find(3)

Topic : Greedy Method

Algorithm Kruskal (E, cost, n, t)
{
1. Construct a heap out of the edge costs using Heapify; (e)
2. for i : = 1 to n do parent[i] := — 1; n
3. i := 0; mincost := 0.0; 1
4 while ((i < n — 1) and (heap not empty)) do e loge
e
{
a) Delete a minimum cost edge (u, v) from the heap
and reheapify using Adjust; loge
b) J = Find(u); k = Find(v);. 1
c) if (j ≠ k) then 1
{
i := i + 1; 1
t[i, 1] := u; t[i, 2] := v; 1
mincost := mincost + cost[u, v]; 1
Union (j, k); 1
}
}
if (i ≠ n –1) then write ("No spanning tree");
else return mincost;
}

Time = c + e + n + e loge

$$\boxed{O(e \cdot \log e)}$$

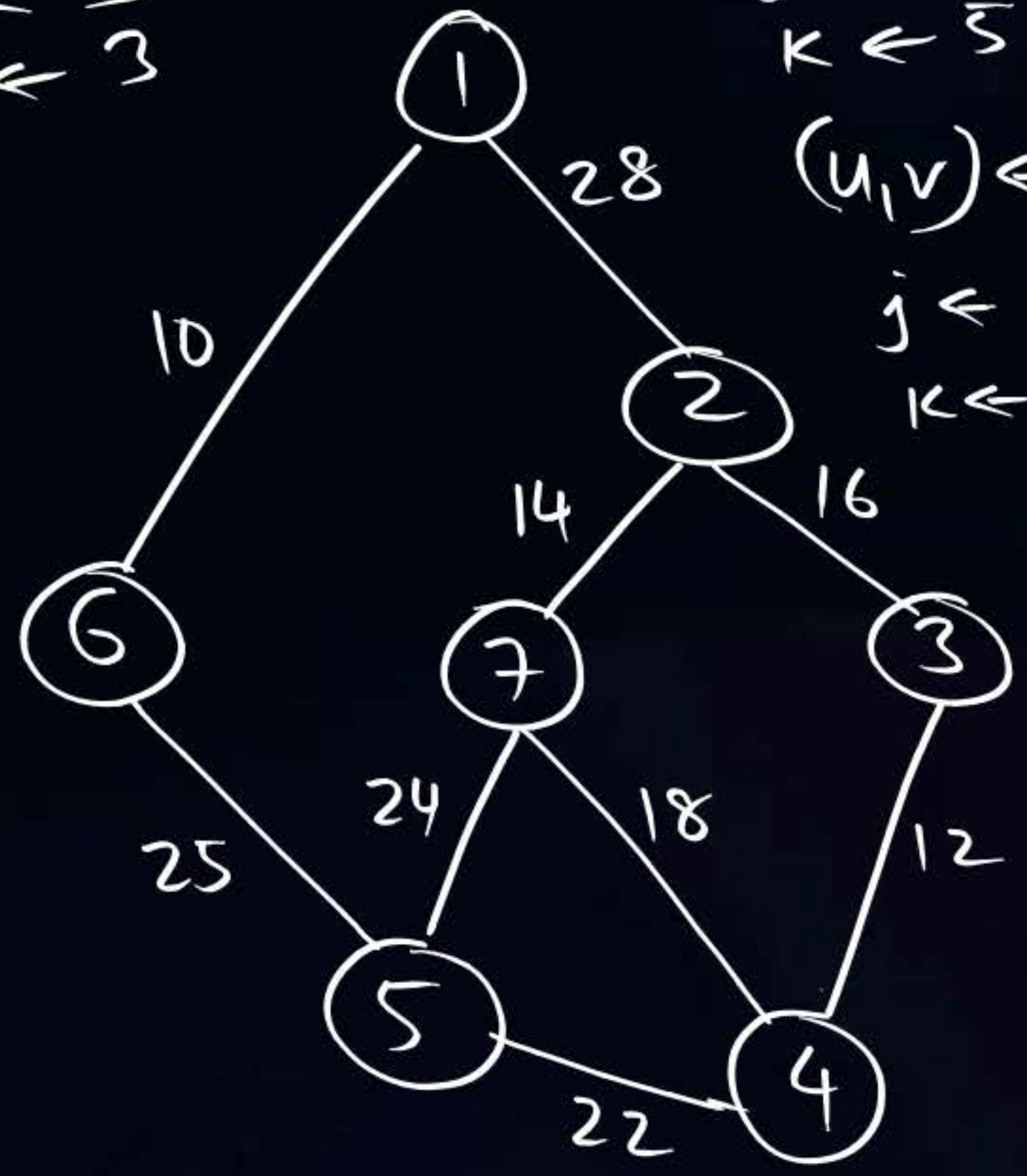
For complete Graph $O(n^2 \cdot \log n)$

Slide 10

$(u,v) \leftarrow (5,7)$
$j \leftarrow 3$
$k \leftarrow 3$

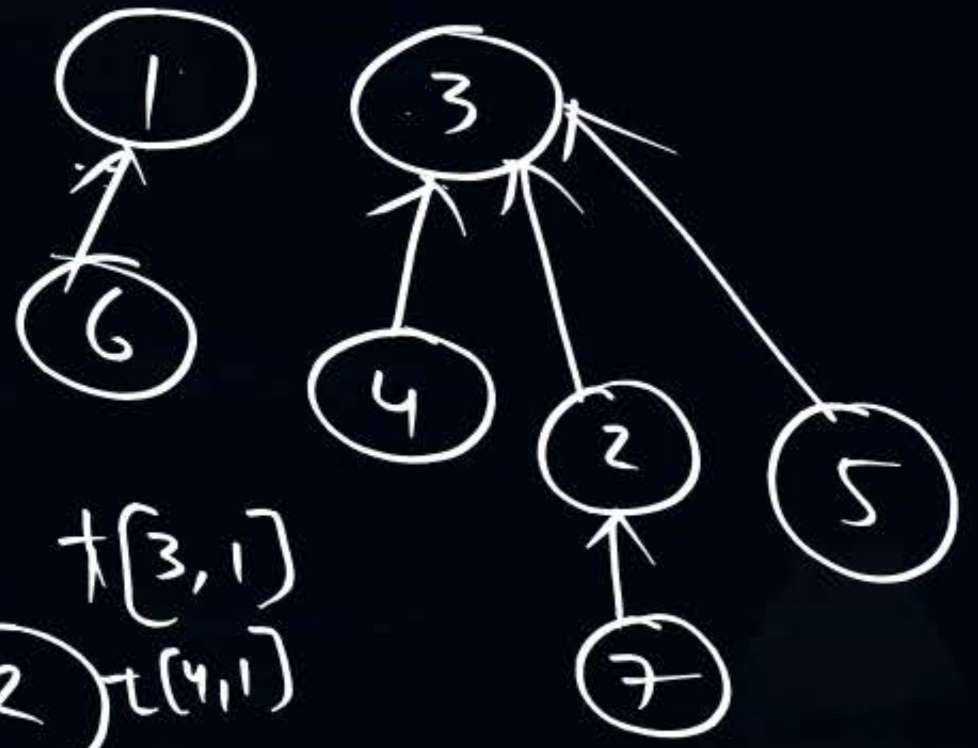
$(u,v) \leftarrow (4,5)$
$j \leftarrow 3$
$k \leftarrow 5$

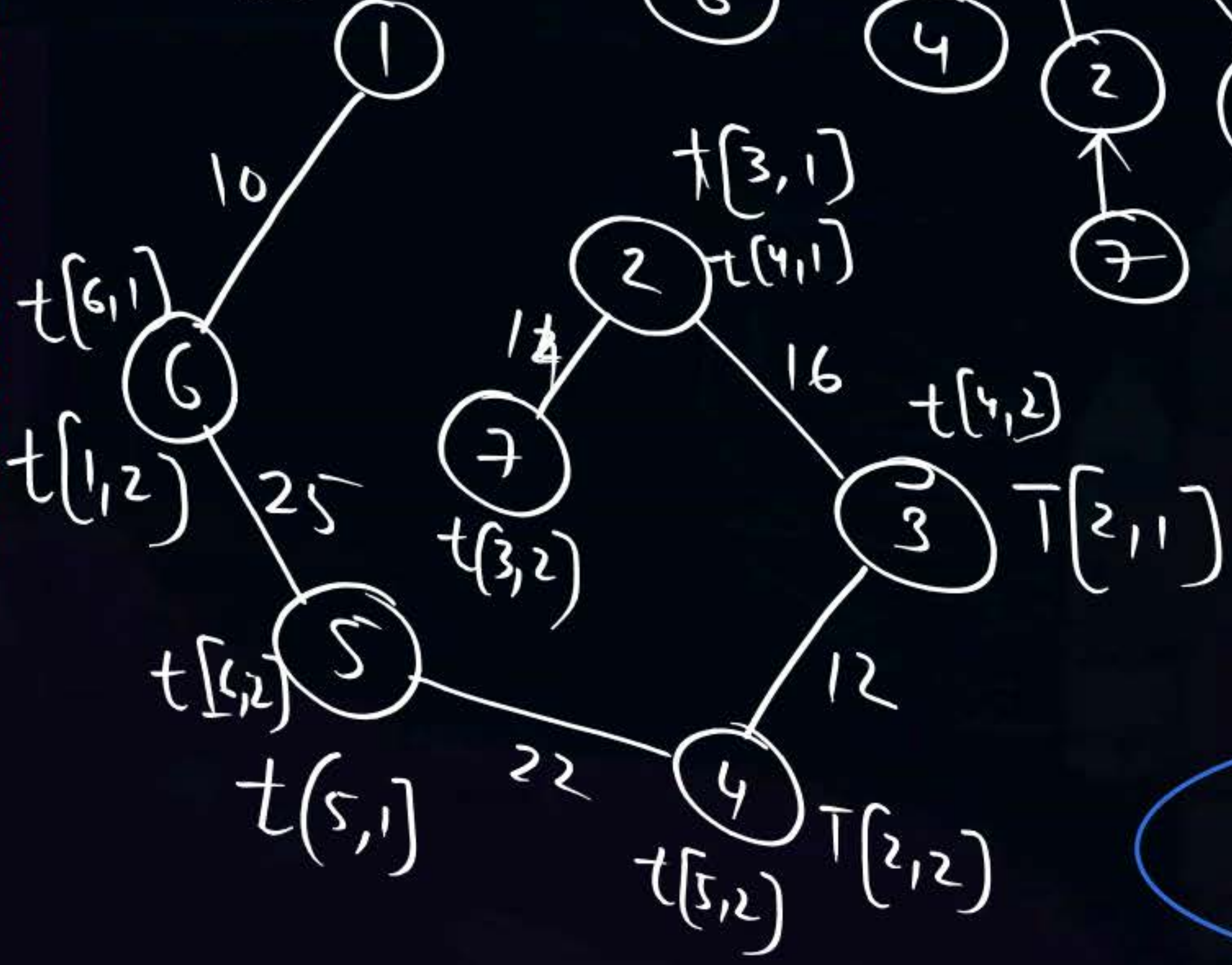
$(u,v) \leftarrow (5,6)$
$j \leftarrow 3$
$k \leftarrow 1$



$\langle u,v \rangle = \langle 1,6 \rangle$
$j \leftarrow 1$
$k \leftarrow 6$

1. Min-Heap : ~~10~~ ; ~~12~~ ; ~~14~~ ; ~~16~~ ; ~~18~~ ; ~~22~~ ; ~~24~~ ; ~~25~~ ; 28





$(u,v) \leftarrow (2,3)$
$j \leftarrow 2$
$k \leftarrow 3$

$(u,v) \leftarrow (2,7)$
$j \leftarrow 2$
$k \leftarrow 7$

$(u,v) \leftarrow (3,4)$
$j \leftarrow 3$
$k \leftarrow 4$

$(u,v) \leftarrow (4,7)$
$j \leftarrow 3 ; k \leftarrow 3$

THANK - YOU