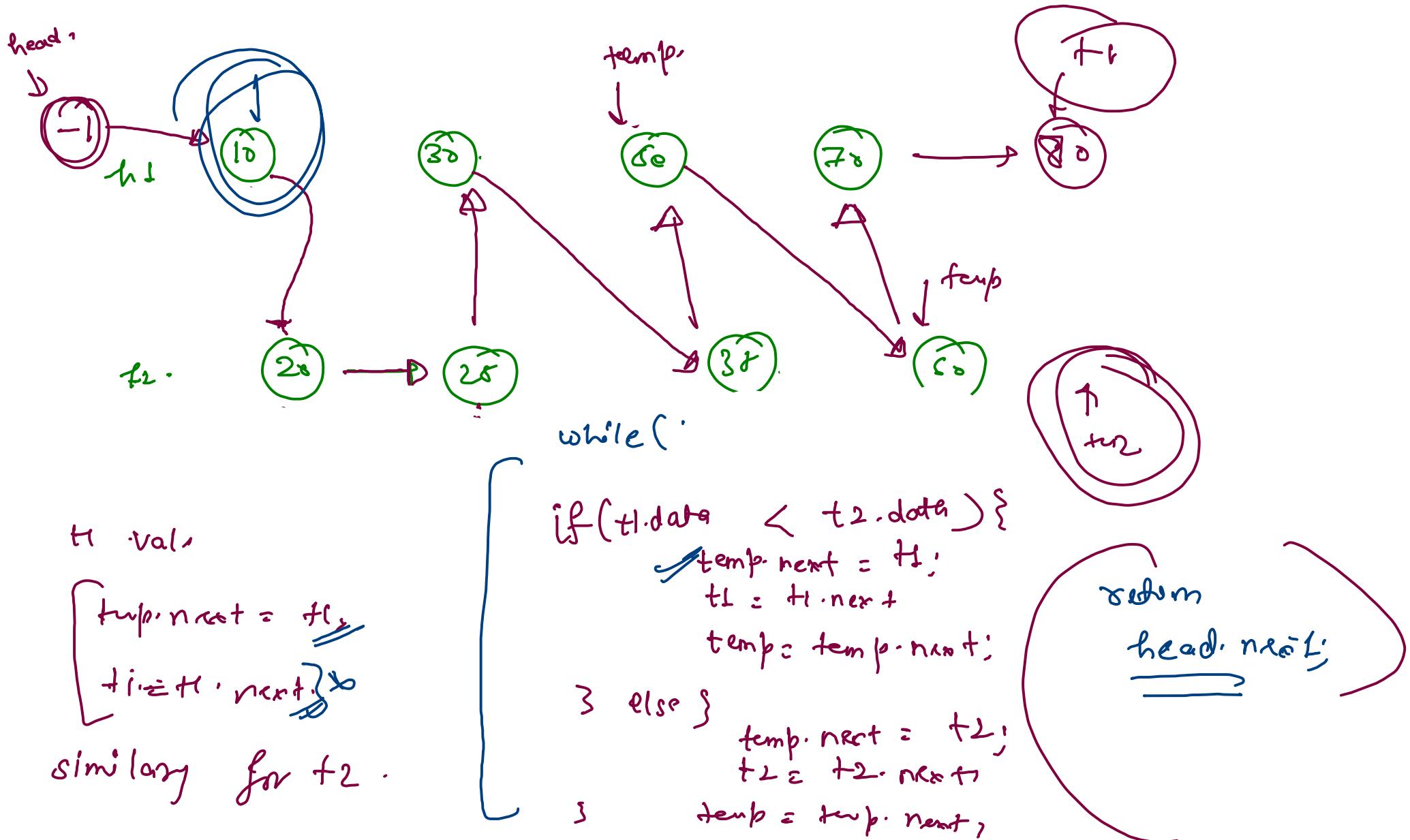
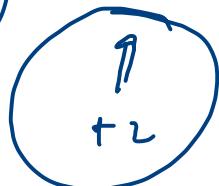
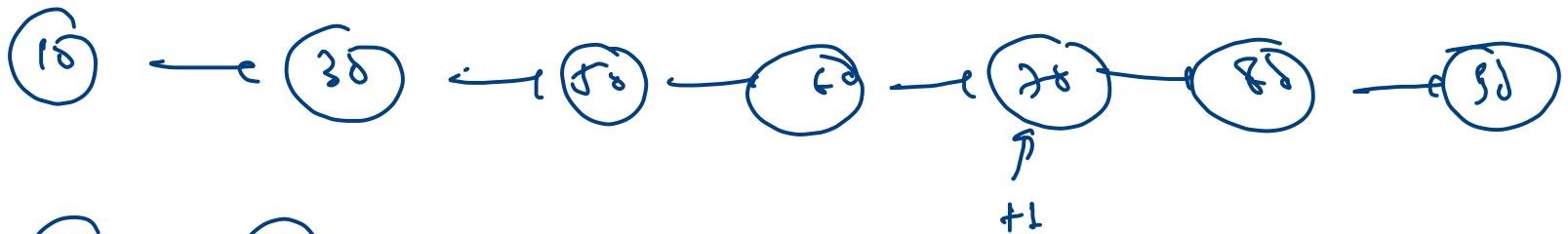


```

if( t1.data < t2.data ) {
  ListNode nn = new ListNode( t1.data );
  temp.next = nn; t1 = t1.next;
} else {
  ListNode nn = new ListNode( t2.data );
  temp.next = nn; t2 = t2.next;
}
temp = temp.next;
  
```

return head.next;





~~if ($t_1 \neq \text{null}$) {~~

~~t1 = t1.next;~~

~~if ($t_1 \neq \text{null}$) {~~

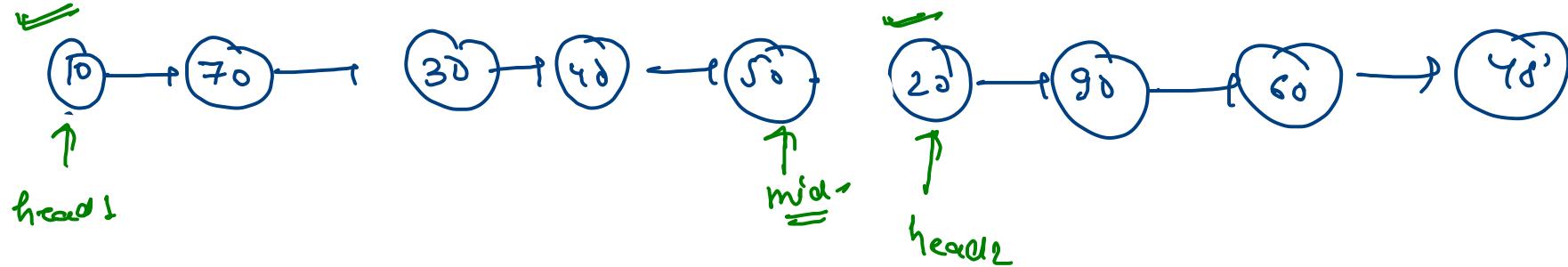
~~t1 = t1.next;~~

>

~~if ($t_2 \neq \text{null}$) {~~

~~t2 = t2.next;~~

$n \log n$ \leq $n \log n$



getMid:

$head_2 = mid.next$

$mid.next = null$

$head_1 = mergeSort(head_1);$

$head_2 = mergeSort(head_2);$

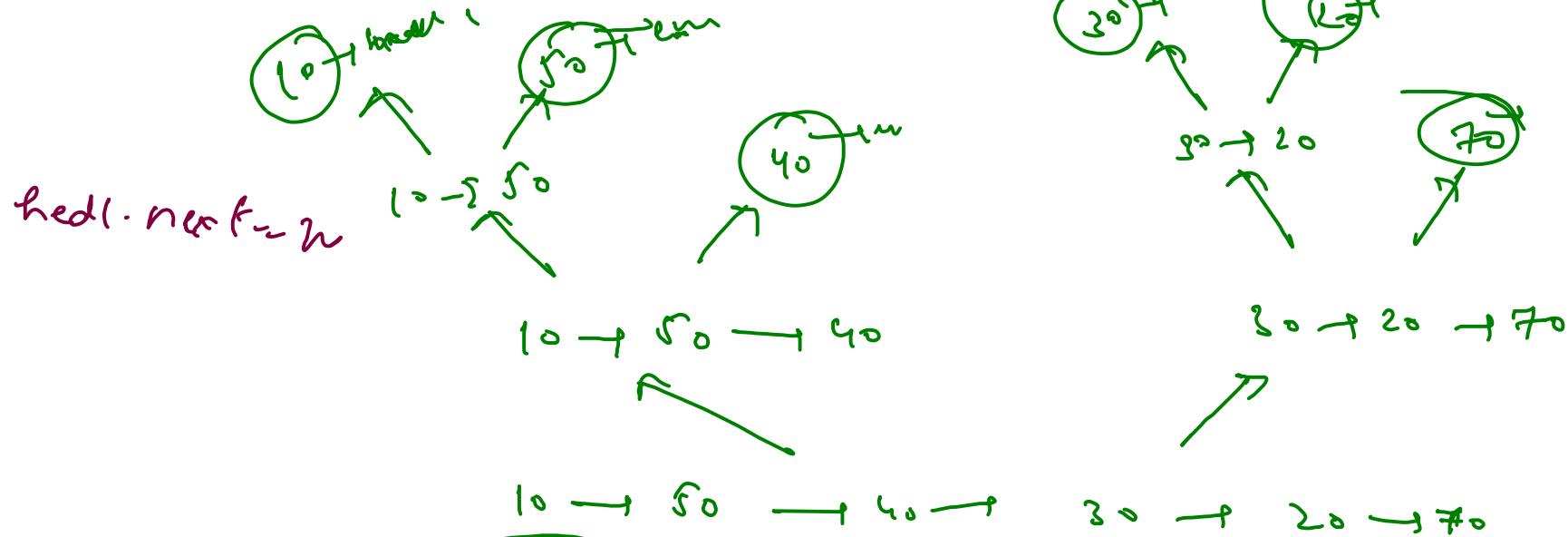
$head_1 = myTwoSorted(head_1, head_2)$

return $head_1$;

$lo - mid \rightarrow arr_1$

$mid + 1 - hi \rightarrow arr_2$

$myMerge(arr_1, arr_2) \rightarrow result$



```
public static ListNode mergeSort(ListNode head1) {
    ListNode mid = midNode(head1);
    ListNode head2 = mid.next;
    mid.next = null;

    head1 = mergeSort(head1);
    head2 = mergeSort(head2);

    head1 = mergeTwoLists(head1, head2);

    return head1;
}
```

~~Boiling~~

$$T(n) = \frac{n}{2} + \underbrace{T\left(\frac{n}{2}\right)}_{2T\left(\frac{n}{2}\right)} + T\left(\frac{n}{2}\right) + C$$

$$T(n) = n + \cancel{2T\left(\frac{n}{2}\right)} + C$$

$$\cancel{2T\left(\frac{n}{2}\right)} = \cancel{\frac{n}{2}} + \cancel{2^2 T\left(\frac{n}{4}\right)} + 2C$$

$$\cancel{2^2 T\left(\frac{n}{4}\right)} = \cancel{\frac{n}{2}} + \cancel{2^3 T\left(\frac{n}{8}\right)} + \cancel{2^3 C}$$

$$\vdots \quad \vdots \quad \vdots$$

$$\cancel{2^{x-1} T\left(\frac{n}{2^{x-1}}\right)} + \cancel{2^{x-1} \frac{n}{2^{x-1}}} + 2^x T\left(\frac{n}{2^x}\right) + 2^x C$$

$$T(n) = n + 2^x T\left(\frac{n}{2^x}\right) + C[1 + 2 + 2^2 + \dots + 2^{x-1}]$$

$$T(L) = T\left(\frac{n}{2^x}\right) \Rightarrow \frac{n}{2^x} = 1, \quad n = 2^x$$

$$\log_2 n = \log_2 2^x = x \log_2 2$$

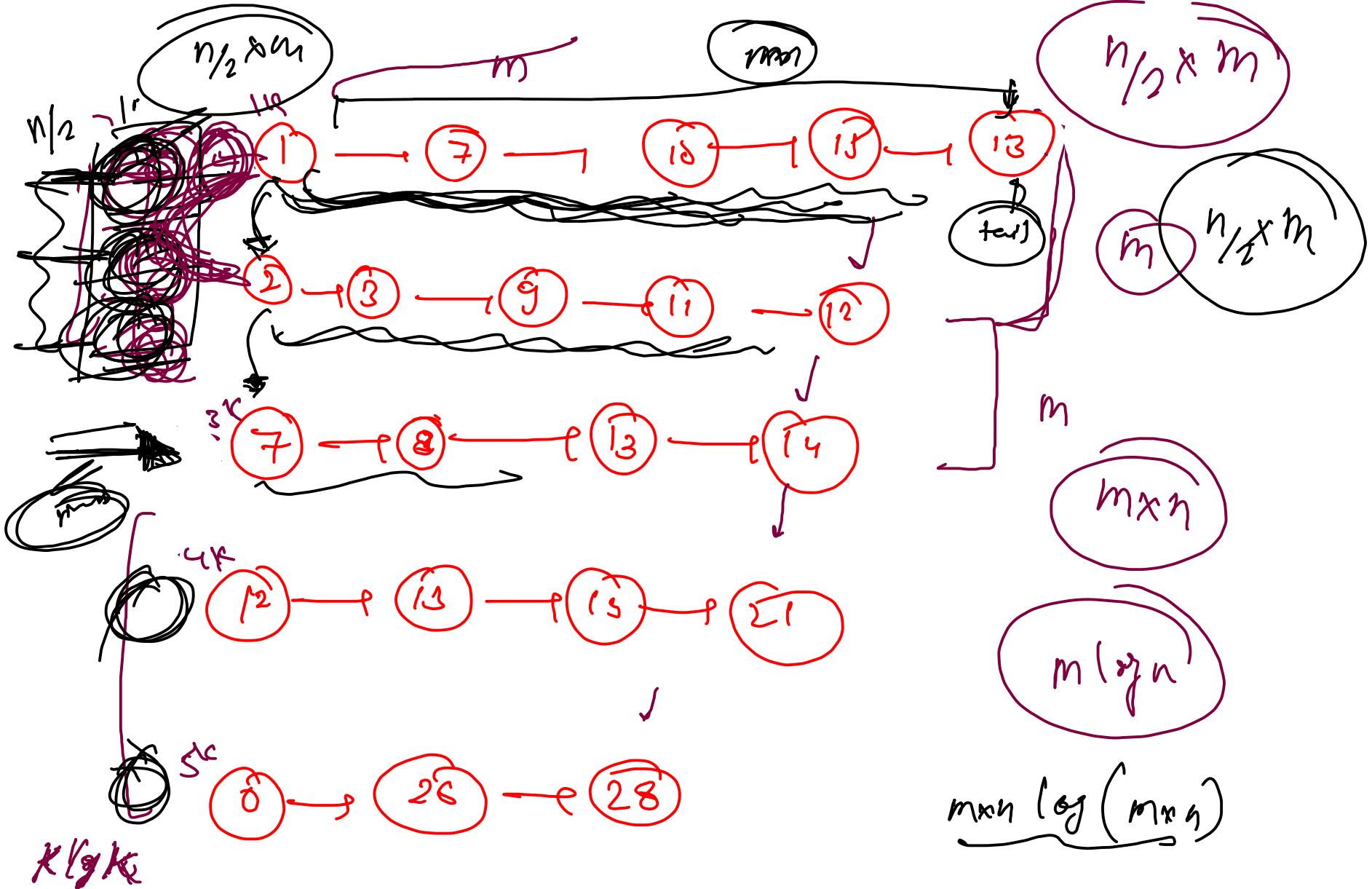
$$\boxed{x = \log_2 n}$$

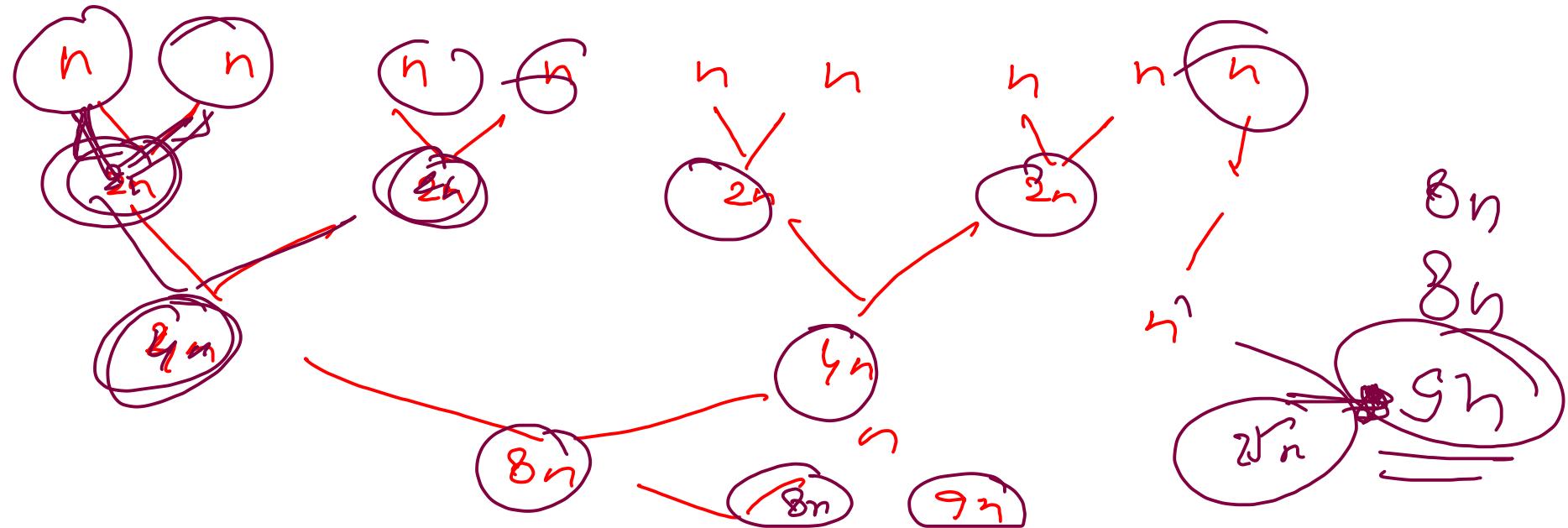
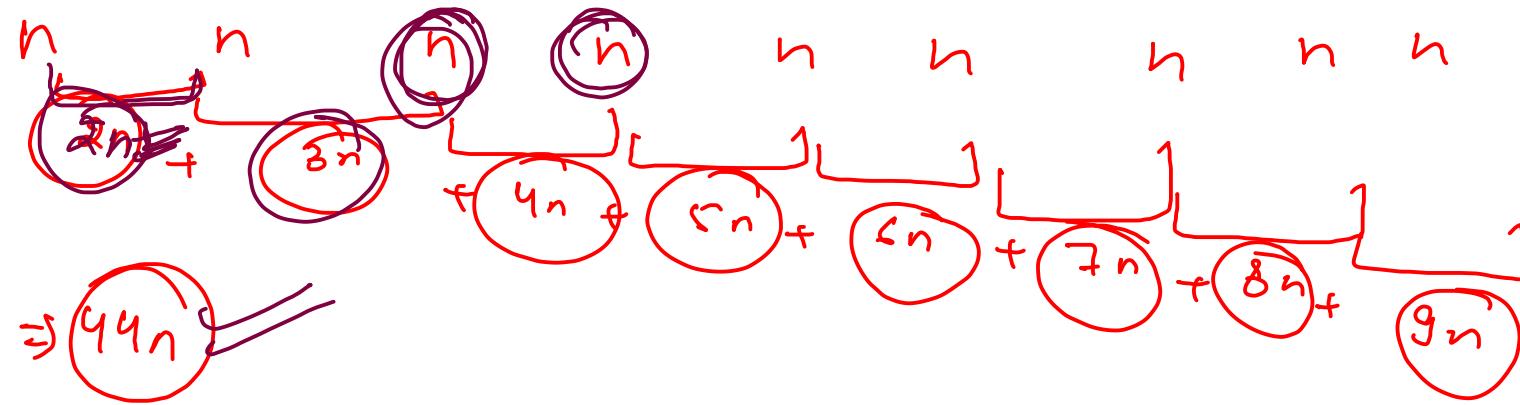
$$T(n) = n \log n + 2^{\log_2 n} x 1 + C[1 + 2 + 2^2 + \dots + \frac{2^x}{2}]$$

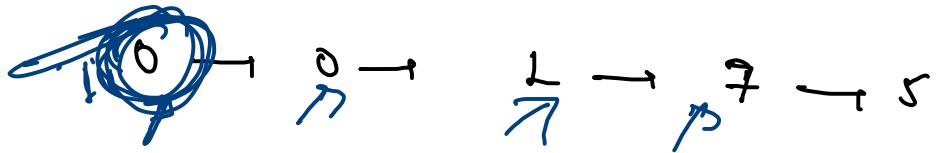
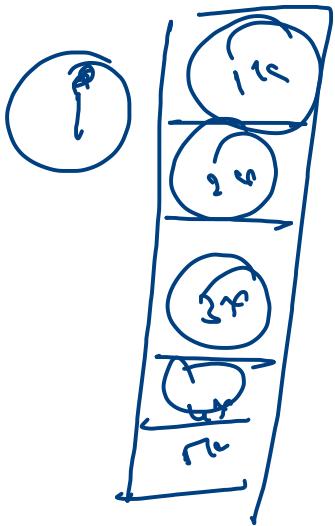
$$T(n) = n \log n + n + C \left[1 \times \frac{(2^x - 1)}{2^x} \right]$$

$$T(n) = n \log n + n + C \left(\underbrace{2^{\log_2 n} - 1}_{L} \right) =$$

$$= \circled{n \log n} + n(C_1) + C_2 \frac{C(n+1)}{\approx} \quad O(n \log n)$$

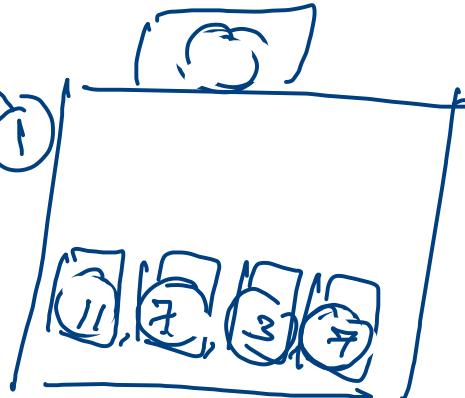


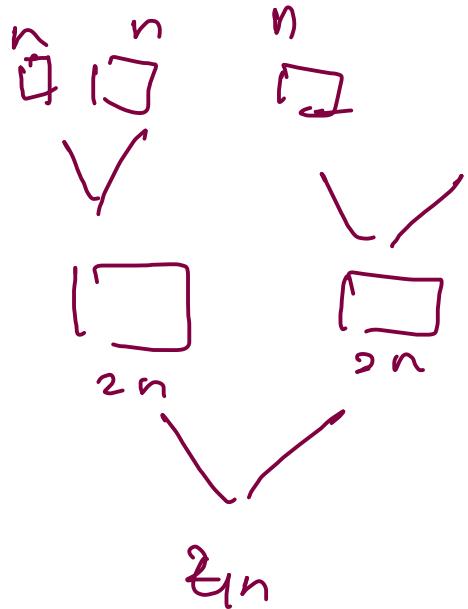




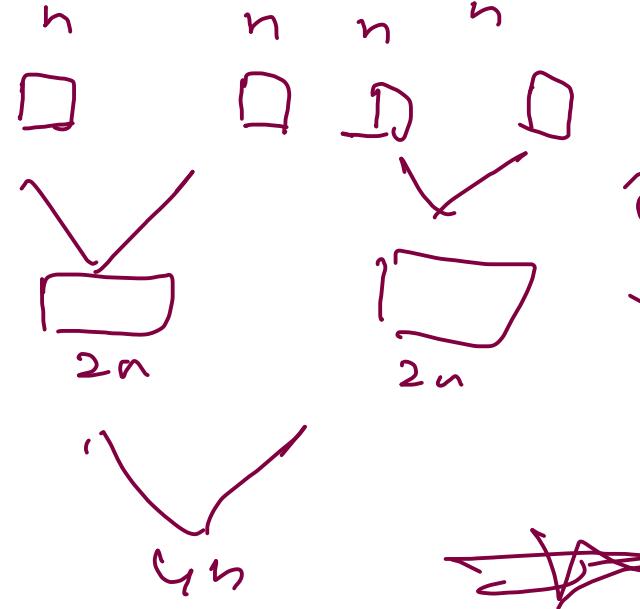
\downarrow
temp.

\downarrow
temp





$N = \text{total No. of Elements}$



$n k \log k$

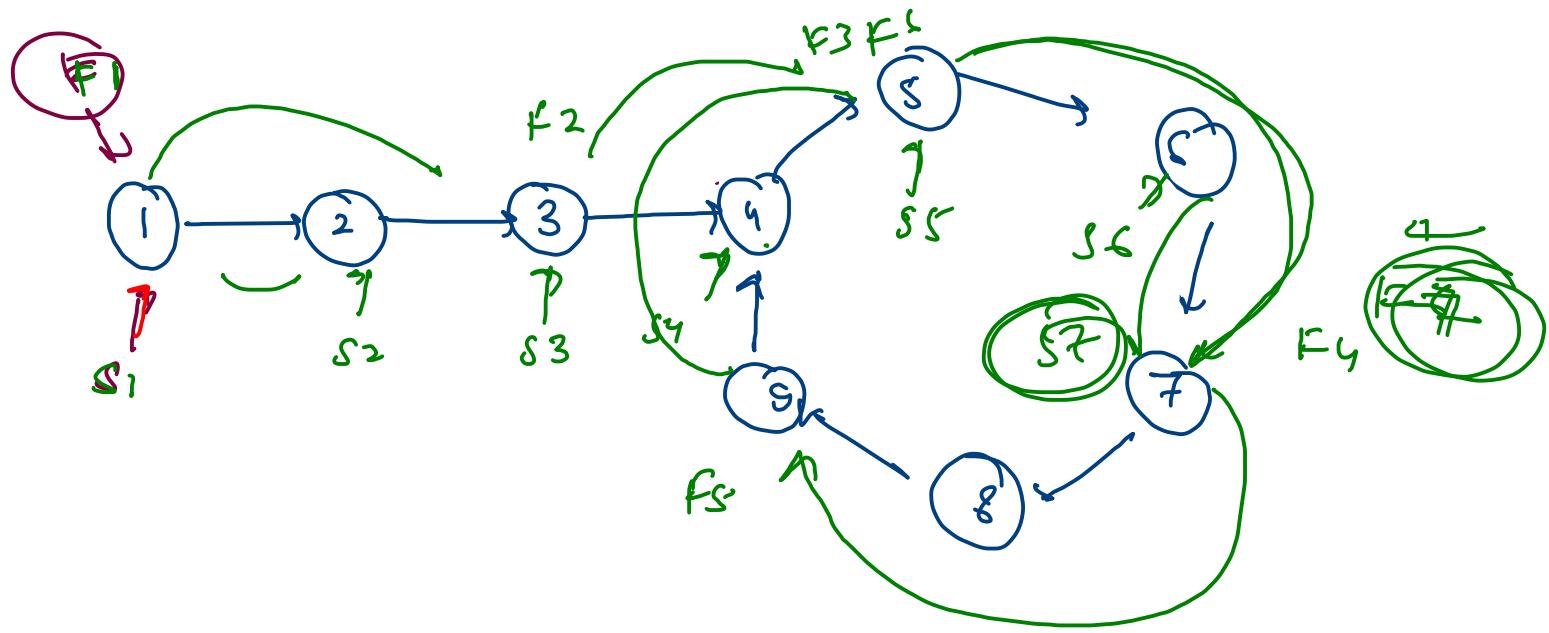
$$n k = N$$

Total No. of
Level $\sim \log k$

$$\left\{ \frac{k}{2} \times 2^n = kn \right.$$

$$\left\{ \frac{k}{2} \times 2^n = kn \right.$$

$$= n k \log k$$



Slow = head;

fast = head.next;

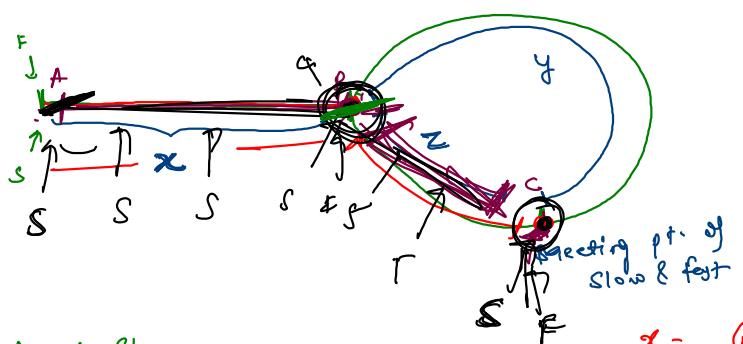
while fast != null && fast.next != null) {

```

if (Slow == fast) {
    Morris break;
}
{Slow = Slow.next}
{Fast = Fast.next}
}

```

SlowSlow.next;
Fast = fast.next.next;
if (Slow == Fast) {
 Morris break;
}



speed of Slow = x

speed of fast = $2x$.

time for reaching at pt C \rightarrow

distance for Slow

$$s = \frac{y}{t}$$

$$t = \frac{x+y}{s}$$

$$\frac{x+y}{s} = \frac{x + m(y+z) + y}{2s}$$

$$2x+2y - x - y = m(y+z)$$

$$x + y = m(y+z)$$

$$x = \underline{my} + \underline{mz} - \underline{y}$$

$$x = \underline{y(m-1)} + \underline{mz} + z - \underline{z}$$

$$x = y(m-1) + z(m-1) + z$$

$$x = (m-1)(y+z) + z$$

Meeting pt. of
Slow & fast

$$x = (m-1)(y+z) + z$$

displacement

\curvearrowleft

\curvearrowright

\curvearrowleft

\curvearrowright

$$\text{Slow} = t \quad x =$$

$$\text{Fast} = t \quad x =$$

$$\text{distance for Fast}$$

$$2s = x + m(y+z) + y$$

$$t = \frac{x + m(y+z) + y}{2s}$$

\curvearrowleft

\curvearrowright

\curvearrowleft

\curvearrowright

\curvearrowleft

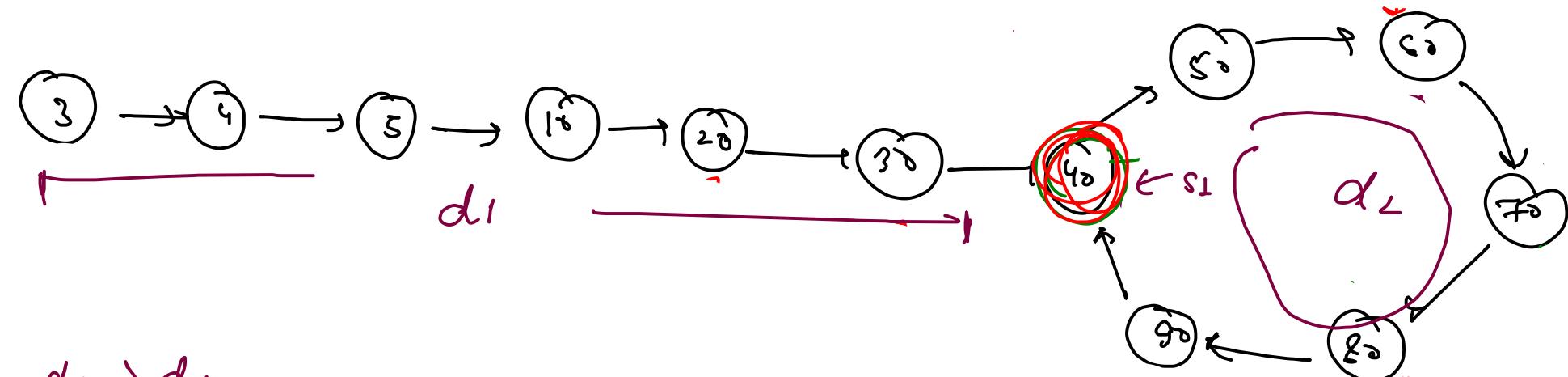
\curvearrowright

\curvearrowleft

\curvearrowright

\curvearrowleft

\curvearrowright



$$\underline{d_1 > d_2}$$

$$\underline{\underline{d_2 > d_1}}$$