Q

Paypal assessment

Given N jobs ⇒ Start & End time

	0	1	2	3
Start :	1	5	7	1
End :	7	8	8	8

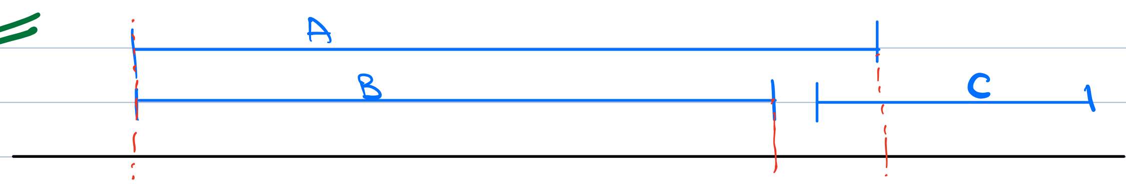
⇒ 2

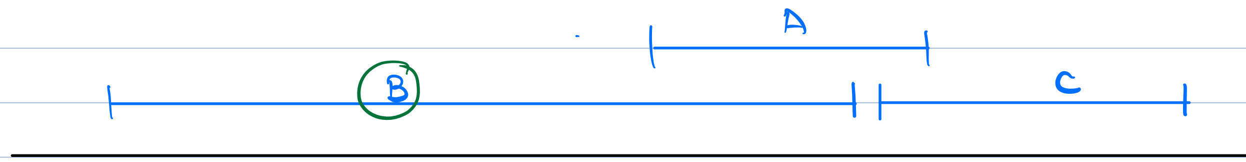
Goal : Perform max jobs possible

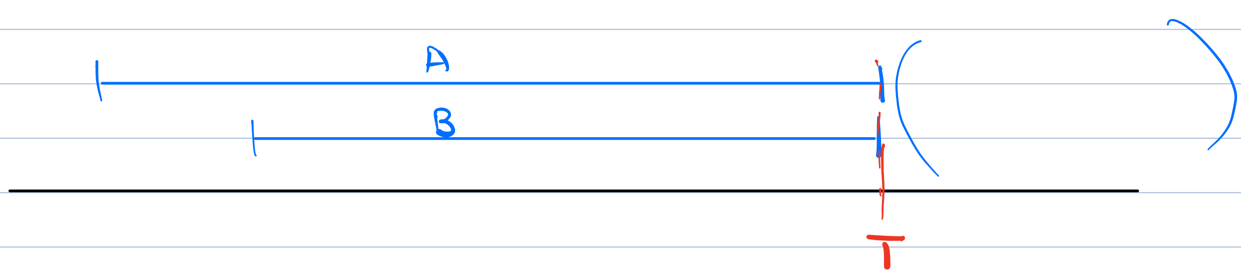
Const: Only perform one job at a time

If a job is ending at T, you can start another job at T.

Solu



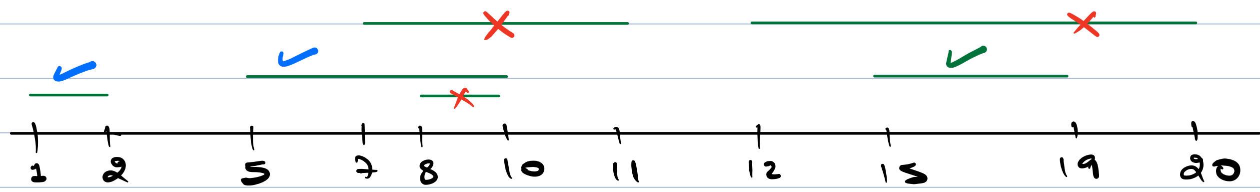




⇒ Start time & duration is not imp.

⇒ We select the job with min End time

	0	1	2	3	4	5
S :	1	5	8	7	15	12
E :	2	10	10	11	19	20



Code

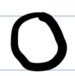
1) Sort the jobs on the basis of End time.

```
int count = 0;
int lastEndTime = 0;

count++;
lastEndTime = E[0];

for (i = 1; i < N; i++) {
        if (S[i] >= lastEndTime) {
                count++;
                lastEndTime = E[i];
        }
}

return count;
```

$$T.C. = O(N\log N + N)$$
$$= O(N \log N)$$

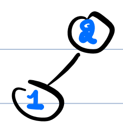
Q// Given N distinct integers.
Find the no. of max heaps possible.

$N = 1$



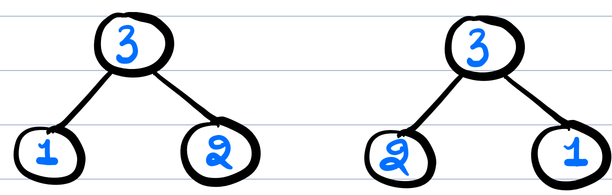
Ans = 1

$N = 2$ (1, 2)



Ans = 1

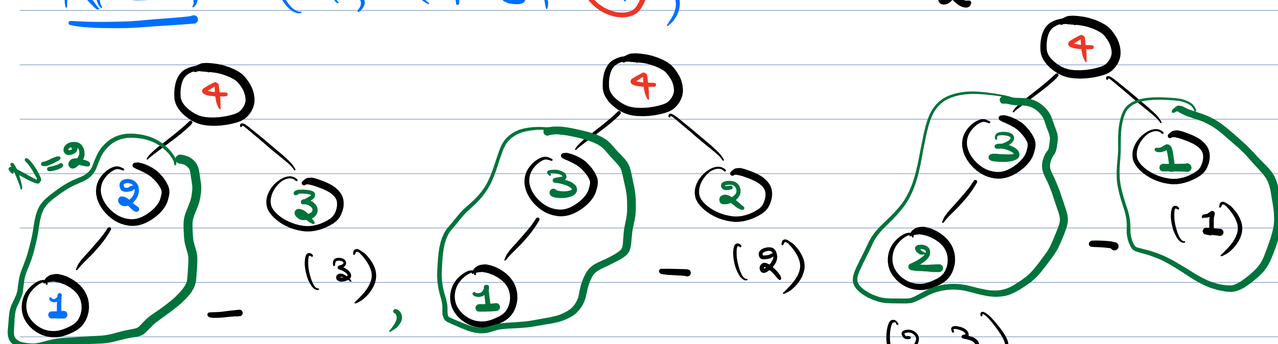
$N = 3$ (1, 2, 3)



Ans = 2

$N = 4$ (1, 2, 3, 4) root

$^3C_2 = 2$



(1, 2) , (1, 3) (2, 3)

$(1) + (1) + (1)$

$= 3 = 3 \times 1 \Rightarrow$

??

Observations

1) Structure of Heap for a given N will be fixed

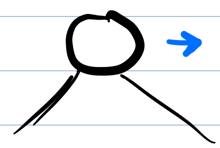
2) Root is also fixed (including subtrees)

3) No relation b/w LST & RST.

   i.e. we have to calculate all possibilities.

N nodes [1 , N]

Root → N



LST   RST

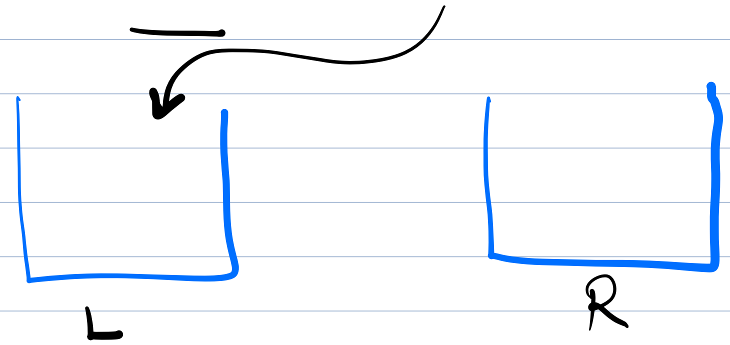
\# LST = L   \# RST = R

$$L + R + 1 = N$$

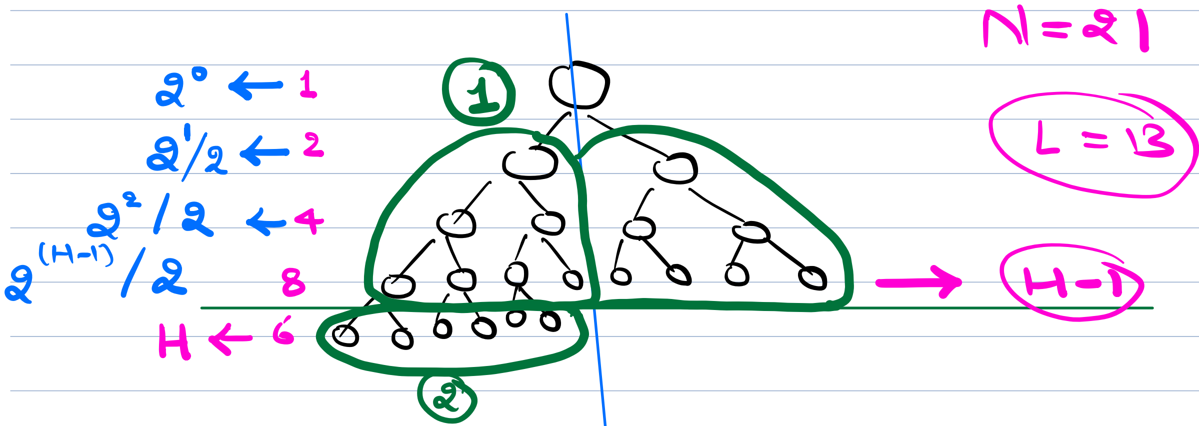
$$L + R = N - 1 \quad \text{—} \quad (1)$$

$$L \geq R \quad \text{—} \quad (2)$$

$N-1 \Rightarrow {}^{N-1}C_L$



Target : Calculate $\underline{\underline{L}}$



Height $= \log_2 N$

$\Rightarrow$ # elements at the i^{th} level $= 2^i$

$\Rightarrow$ Last level might not be completely filled.

$\Rightarrow$ Except the last level, elements are equally divided b/w LST & RST.

1) # Elements in LST in all levels except the last level.

Total no. of elements in all levels are last.

$$= \overset{a}{2^0} + 2^1 + 2^2 + 2^3 \ldots\ldots 2^{H-1}$$

$r = 2, \quad N = H$

$$= \frac{1(2^H - 1)}{2-1} = 2^H - 1 \quad — (3)$$

Elements in the last level

$$N - (2^H - 1) \quad — (4)$$

Max no. of nodes in last level.

$$= 2^H$$

Max no. of elements in the last level in LST.

$$\frac{2^H}{2} = 2^{H-1}$$

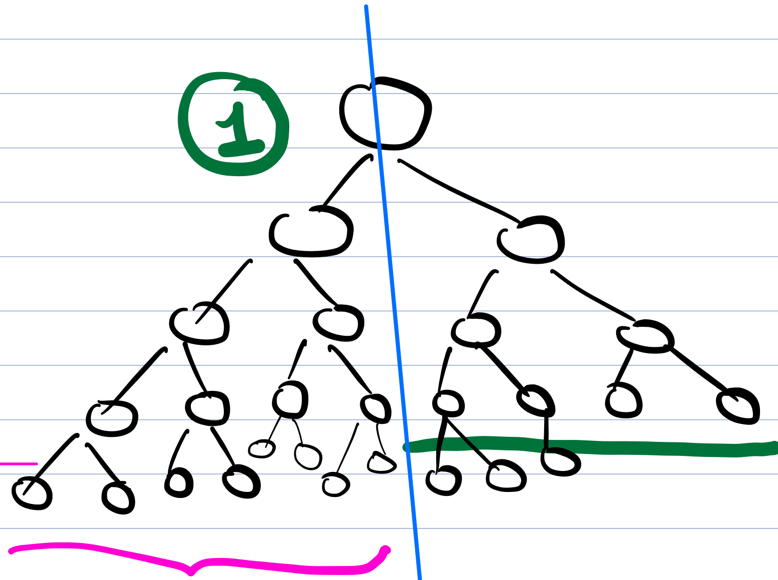
$2^0 \leftarrow 1$

$2^1/2 \leftarrow 2$

$2^2/2 \leftarrow 4$

$2^{(H-1)}/2 \quad 8$

~~16~~ 8



max = 8

Total no. of elements in last level.	max	# Elements in LST in last level
4	8	4
5	8	5
6	8	6
7	8	7
8	8	8
9	8	8
10	8	8
11	8	8

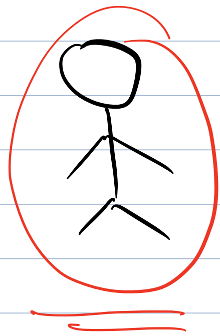
max

$$\text{\# Elements in the LST in last level} = \min\left(2^{H-1},\ N-(2^{H}-1)\right)$$

$$\boxed{L = \min\left(2^{H-1},\ N-(2^H - 1)\right) + \frac{\left(2^H - 1 - 1\right)}{2} = \min\left(\frac{x+1}{2},\ N-x\right) + \left(\frac{x-1}{2}\right)}$$

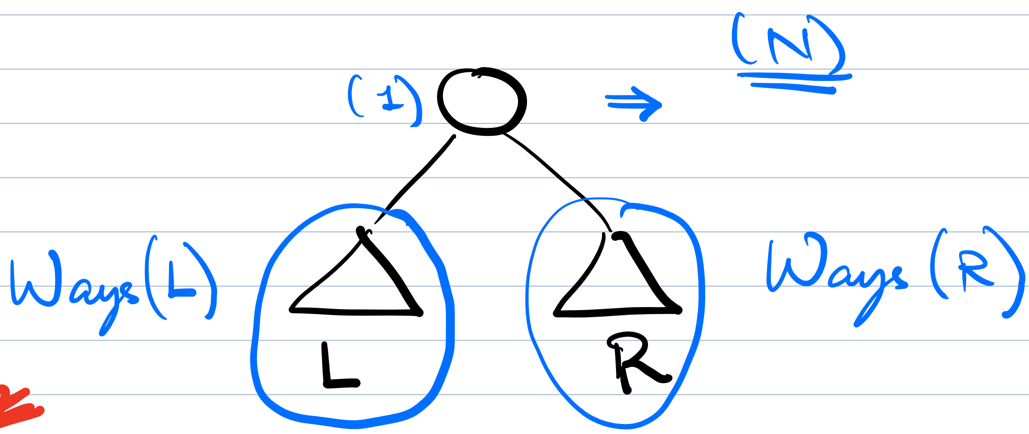
$$\frac{1+x}{2} = 2^H/2 = 2^{H-1}$$

(S, J)



5 Shirt

3 Jeans.



$$\boxed{\text{Ways}(N) = \left({}^{N-1}C_L\right) \times \left(\text{Ways}(L) \times \text{Ways}(R)\right)}$$

Distributing $(N-1)$ into L & R

Code

```
int Ways (N) {
    if (N<=2) { return 1; }
    int H = log_2 N
    int x = (2^H - 1)
    int L = min(N-x, (x+1)/2) + (x-1)/2
    int R = (N-1) - L
    int ans = (N-1 C L) x Ways(L) x
                           Ways(R);

    return ans;
}
```

Comparator

1st → int a, 2nd → int b

bool function (int a, int b)

True

false