

Today's Content : 43 length

1. SORAT

2. Painten Parbhim.

3Q: Given +ve find SQR(N)

Find greatest i such that $i^2 \leq N$

Sqrt(25) : $5^2 \leq 25$

Sqrt(37) : $6^2 \leq 37$ $7^2 > 37$

N = 30

i	$i^2 \leq N$	ans
1	$1^2 \leq 30$	ans = 1
2	$2^2 \leq 30$	ans = 2
3	$3^2 \leq 30$	ans = 3
4	$4^2 \leq 30$	ans = 4
5	$5^2 \leq 30$	ans = 5
6	$6^2 \leq 30$	return ans = 5.

```
int sqrt(int N) { TC: O(√N) SC: O(1)
```

```
    int i = 1, ans = 0;  $i^2 \leq N \Rightarrow i \leq \sqrt{N}$ 
```

```
    while(i^2 <= N) { i = 1..√N : √N iterations
```

```
        ans = i;
```

```
    } i++;
```

```
    return ans;
```

```
}
```

2. Idea:

1. Target: Greatest ele, $ele^2 \leq k$

2. SearchSpace: [Where we search our ans: Answer space]
[Note: This can be real or imaginary]

Ex: SearchSpace: [1...N], Total Elements =

3. Discard?

N = 30 SearchSpace []

l	h	m
1	30	15
1	14	7
1	6	3
4	6	5
6	6	6
6	5	

$15^2 \leq 30$: No $15^2 > 30$ go left: $h = m - 1$

$7^2 \leq 30$: No $7^2 > 30$ go left: $h = m - 1$

$3^2 \leq 30$: Yes ans = 3; $1 \ 2 \ 3$ go right: $l = m + 1$

$5^2 \leq 30$: Yes ans = 5; $4 \ 5$ go right: $l = m + 1$

$6^2 \leq 30$: No go left: $h = m - 1$

$6 > 5$: return ans = 5.

Constraints:

$$1 \leq N \leq 10^9$$

```
int sqrt(int N) {
```

Note: Search Space Size $\{1..N\} = N$ Elements

```
int l=1, h=N, ans=0;
```

TC: $O(\log_2 N)$ SC: $O(1)$

```
while (l <= h)
```

↳ Binary Search Iterations.

```
int m = (l+h)/2;
```

```
if (m*m <= N) {
```

```
ans = m;
```

```
} l = m+1; // goto right
```

```
else {
```

```
h = m-1; // goto left
```

```
}
```

```
return ans;
```

```
}
```

Q1:

We have to paint N boards of length $\{c_0, c_1, c_2, \dots, c_{N-1}\}$

There are W painters available and each of them takes 1 unit of time to paint 1 unit of board

Calculate & return the minimum time required to paint all boards

Notes:

1. Two painters cannot share a board to paint, A board cannot be painted partially by 1 painter & partially by another.

7 10

2. A painter can only paint contiguous boards

5 10

3. All painters start work at same time.

4 10

Ex:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	→ 21	
N=15 :	3	5	1	7	8	2	5	3	10	1	4	7	5	4	6	Time taken	
W=3 :	W ₁ =34								W ₂ =15				W ₃ =22				34
	W ₁ =26					W ₂ =23					W ₃ =22					26	
	W ₁ =24				W ₂ =25							W ₃ =22				25	

Ex2:	0	1	2	3	4	5	→ 106
$arr[6] =$	1	1	1	1	1	101	Time Taken
$W=2$	$W_1=5$				$W_2=101$: 101 mins

Idea: Avg allocation

Total time = 106 $k=2$ avg time = 53

Note: Avg concept won't work.

Let's Search:

Target: Min time required to finish all tasks

SearchSpace: Based on our target \Leftrightarrow [Based on Time]
 $\{l \quad h\}$

{ Smallest time possible to complete } l : Min time taken for a single task

{ largest time possible to complete } h : Sum of time taken for all tasks

Discard?	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14		
N = 15:	3	5	1	7	8	2	5	3	10	1	4	7	5	4	6		
W = 4:	W ₁ = 9			W ₂ = 15		W ₃ = 10			W ₄ = 15			Tasks are left out. *					
	W ₁ = 34								W ₂ = 37							W ₃	W ₄

SearchSpace: $\{10 \quad 71\}$ // on total time taken to finish all tasks

En: Say mid lands at = 40 Can we finish all tasks by 40 min using 4 workers

Note: If we can do a task in 40 mins
 $\boxed{40} \quad 41 \quad 42 \quad 43 \dots$ We can do same in 41, 42, 43. -

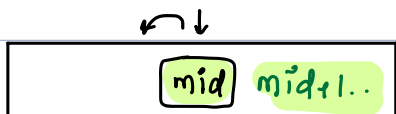
ans = 40;

go left for a better ans.

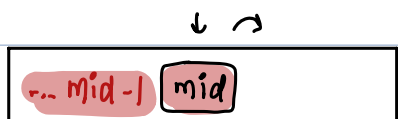
En: Say mid lands at = 15 Can we finish all tasks by 15 min? using 4 workers

$12 \quad 13 \quad 14 \quad \boxed{15}$ Note: If we cannot do a task in 15 mins
go right We cannot do same in 14 13 12...

Discard:



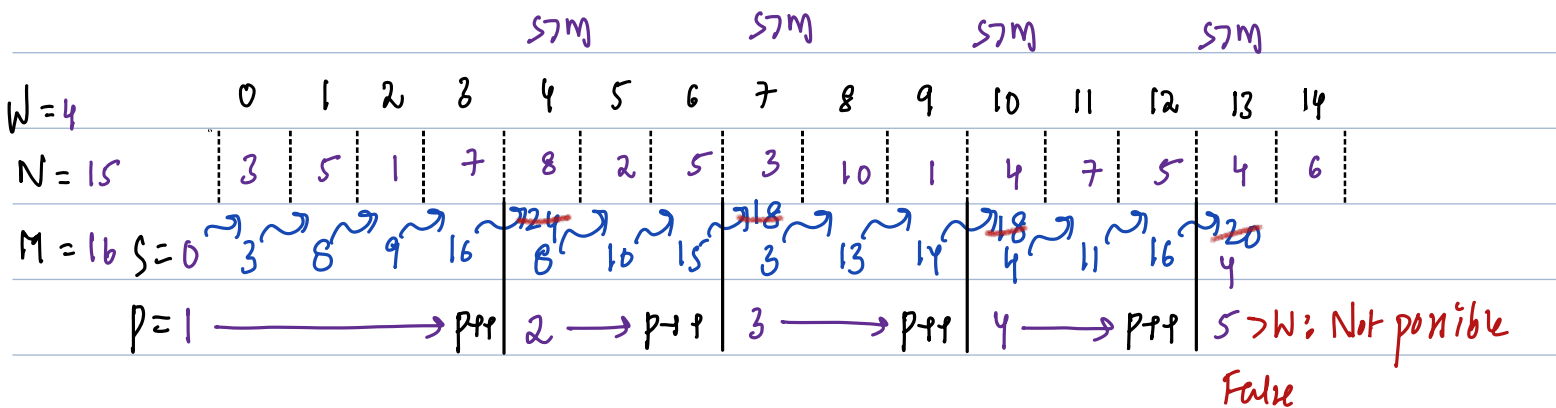
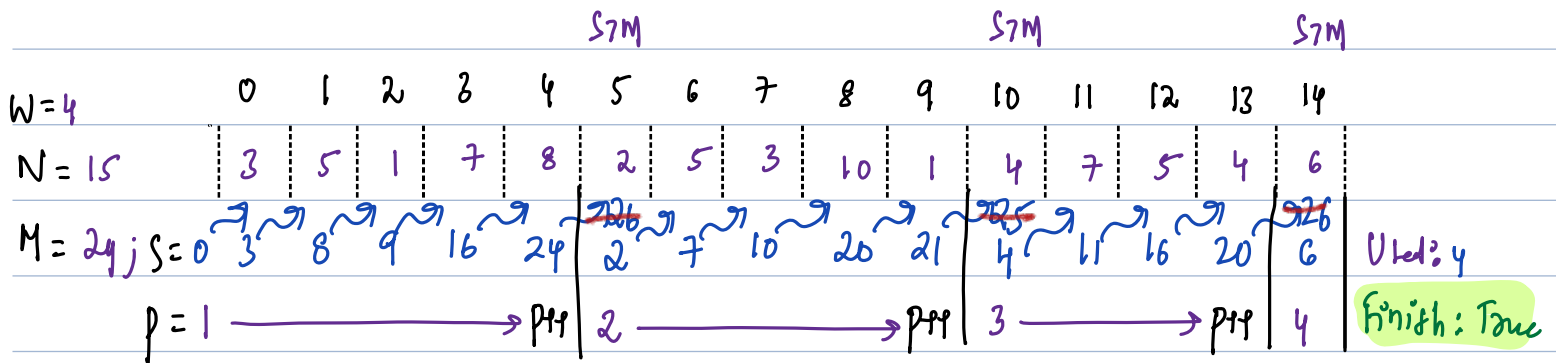
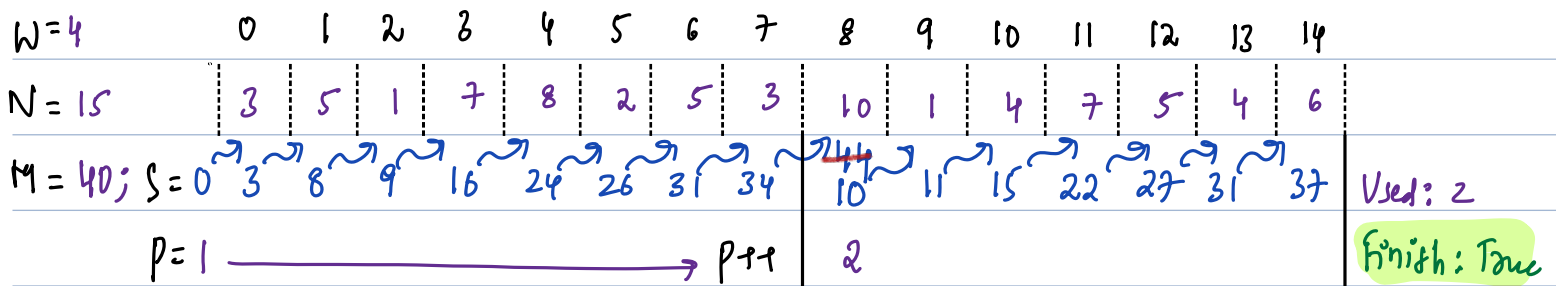
if (finish task in mid time) { ans = mid; $h = m-1$ }



if (Cannot finish task in mid time) { $l = m+1$ }

Check function Idea:

S7M



check part?

l	h	m: l+h/2	Can we finish tasks in m time	Update
10	41	40	Can we paint all boards in 40?	ans = 40; h = m-1;
10	39	24	Can we paint all boards in 24?	ans = 24; h = m-1;
10	23	16	Can we paint all boards in 16?	l = m+1;

Continue Todo;

check func:

class solution {

Pseudo Code boards workers length of each board : unit length = 1 time point

int minTime(int N, int W, int length[]) { Note:

int l = min of length[], h = sum of length[], ans = h;

while (l <= h) {

// Search Space: { l..h } = $h - l + 1$

int m = (l+h)/2

// B.S iterations = \log_2^{h-l+1}

// Check if we can finish N tasks in m time using W workers

if (check(N, W, length, m)) {

Overall TC = $\log_2^{h-l+1} \times N$

ans = m;

SC = O(1)

h = m-1;

else {

l = m+1;

}

}

return ans;

limit

boolean check(int N, int W, int[] length, int m) { TC: O(N) SC: O(1)

int p=1, s=0;

for (int i=0; i < N; i++) {

s = s + length[i];

if (s > m) {

p++;

s = length[i];

if (p > W) { return false; }

}

return true;

}

}

When?

1. Search based question, check Binary
2. Calculate min — } check if we can binary
Calculate max — }

Any Binary Search

1. Target :
2. Search Space : { Based on Target, create l, r }
3. Discard or Not?