Bonus Lecture - 3
Sparse Table

RECAP

Range max / min / gcd / lcm / - - -

Introduction

Inefficient Preprocessing

$\rightarrow O(N^2)$

Efficient Querying

$\rightarrow O(1)$

No preprocessing

$\to O(1)$

Inefficient querying

$\to O(N)$

Preprocess answers only for subarrays of length = 1, 2, 4, 8, 16 . . . . . .

Intuition behind $\underbrace{\text{Sparse}}$ Table

Answer for a few important subarrays will be stored.

The magic happens when we realize that any number can be split into sum of powers of 2

25

More detailed Explanation

Preprocessing

$dp[N][\log_2 N + 1] \Rightarrow$ $dp[i][k]$ will represent the answer to the subarray that starts at index i & has a length of 2^k.

$\Downarrow$ $0, 1, \dots N-1$

$\Downarrow$ $2^0, 2^1, 2^2 \dots 2^{\lfloor \log_2 N \rfloor}$

$\Downarrow$

$N * \log_2 N$ dp states

LearnYard

$dp[i][0] \Rightarrow arr[i \ldots i] \quad len = 1$

$dp[i][1] \Rightarrow arr[i \ldots i+1] \quad len = 2$

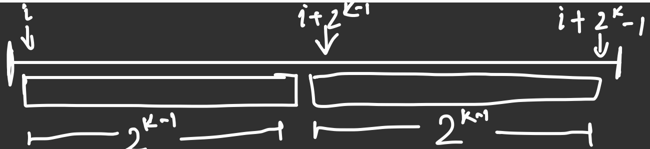
$dp[i][2] \Rightarrow arr[i \ldots i+3] \quad len = 4$

$dp[i][3] \Rightarrow arr[i \ldots i+7] \quad len = 8$

Preprocessing for Range Min

// Base Cases

$$dp[i][0] = arr[i];$$



$$dp[i][k] = \min(dp[i][k-1],\ dp[i+(1<<(k-1))][k-1]);$$

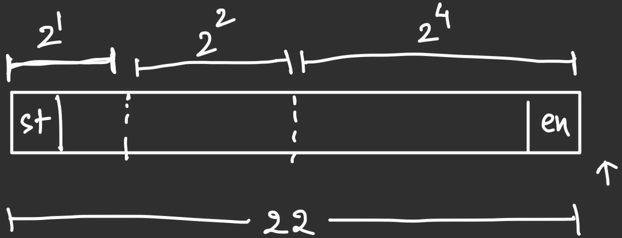
Querying:

$q = [st, en]$ $\quad len = en - st + 1 \Rightarrow 22$

$\Downarrow$

$2^1, 2^2, 2^4$

$\cancel{0}$
1
2
$\cancel{3}$
4



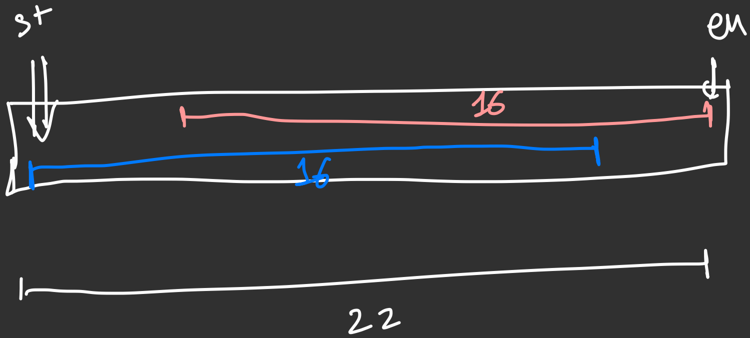
ans $\Rightarrow a_3$, $i = st + 2^1 + 2^2 + 2^4$

Let’s do some problems!

1. Range Min Queries

Let’s implement

Further optimisation for querying



2. Sereja and D

$N = 5$

$[1, 2, 3, 10, 50]$

$\;\;0 \;\; 1 \;\; 2 \;\; 3 \;\; 4$

$t = 1, \quad d = 1 \implies i = 0 \implies 1$

$t = 5, \quad d = 3 \implies i = 0 \implies 1$

$t = 11, \quad d = 7 \implies i = 0 \implies 1$

$t = 100000, \quad d = 1 \implies i = 4 \implies 5$

$t = 10^6, \quad d = 10^6 \implies i = 0 \implies 1$

$t = 11, \quad d = 6 \implies i = 3 \implies 4$

$k \geq i$

1. abs diff for consec. $\leq d$
2. $a[k] \leq t$
3. $a[k+1] > t$

LearnYard

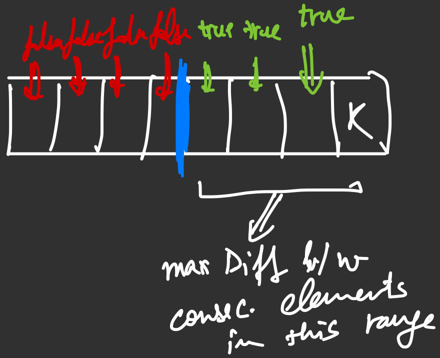
Intuition / Solution

Query $\Rightarrow$ Unique K

$\Downarrow$

Smallest i (s.t. $i \le k$)

$\Rightarrow$ $a[id+1] - a[id] \le d \quad \forall\ id \in [i, k-1]$



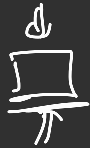
1) Use upper_bound to find accurate K.

2) Use Binary search to find smallest i s.t. $arr[i - - - K]$ is validity.

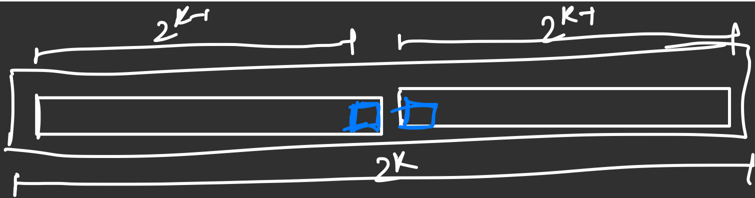
a) Check ⇒ Use sparse Table to find range max Diff & check whether that is $\leq d$ or not?

Preprocessing

len = 1

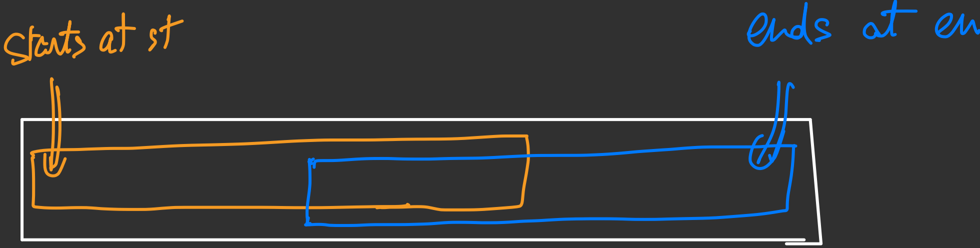


$$dp[i][0] \Rightarrow 0$$



$$dp[i][k] = \max\big(dp[i][k-1],\ dp[i+(1<<(k-1))][k],\ arr[i+(1<<(k-1))] - arr[i+(1<<(k-1))-1]\big)$$

Querying



Largest pow of 2 $\Rightarrow (x) \leq len \Longrightarrow \log_2[len]$

$$2^k > \frac{len}{2}$$

Let’s implement

Thank You!

Reminder: Going to the gym & observing the trainer work out can help you know the right technique, but you'll muscle up only if you lift some weights yourself.

So, PRACTICE, PRACTICE, PRACTICE!