Ternary Search

→ Lets consider a basic problem

Q. Given a sorted array & a target element, find the index of target.

↳ We can solve the above problem using binary search.

1	2	3	4	5	6	7	8	9	10	11	12

1st partn 2nd partn 3rd parti

1

1

9 > 1 True

$n \rightarrow \frac{n}{3}$

DNC

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

$a \rightarrow$ the no. of sub problems in the next level

$b \rightarrow$ the reduction factor of smaller subproblem

$f(n) \rightarrow$ for each subproblem, how much time is spent in worst case

$\hookrightarrow \; T(n) = a\,T\left(\frac{n}{b}\right) + O(n^k \times \log^p n)$

$\searrow f(n)$

Master theorem

1) $\rightarrow$ if $\underbrace{a > b^k}$ then $T(n) = O(n^{\log_b a})$

2) $\rightarrow$ if $a < b^k$

$\rightarrow$ if $p \geq 0 \rightarrow T(n) = O(n^k \log^p n)$

$\rightarrow$ if $p < 0 \rightarrow T(n) = O(n^k)$

3) $\rightarrow$ if $a = b^k$ $\rightarrow p > -1 \rightarrow T(n) \rightarrow O(n^{\log_b a} \times \log^{p+1} n)$

$\rightarrow p = -1 \rightarrow T(n) \rightarrow O(n^{\log_b a} \times \log\log n)$

$\rightarrow p < -1 \rightarrow T(n) \rightarrow O(n^{\log_b a})$

merge sort $\quad a = 2 \quad b = 2$ merge 2 sorted array

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

$$T(n) = aT\left(\frac{n}{b}\right) + O(n^k \log^p_b n)$$

$a = 2 \quad b = 2 \quad k = 1 \quad p = 0$

$b^k \Rightarrow$

$a == b^k$

$$T(n) = O(n^{\log_b a} \times \log^{p+1} n) \Rightarrow O(n^{(\log_2 2)} \times \log n)$$

$$\Rightarrow O(n \log n)$$

Binary → Time Complex

$$T(n) = T(n/2) + O(1)$$

$a=1$ $b=2$ $k=0$ $p=0$

$b^k \Rightarrow 2^0 = 1$

$O(n^{\log_b a} \times \log^{p+1} n)$

$\to O(n^{\log_2 1} \times \log n)$

$O(n^0 \times \log_2 n)$

$\hookrightarrow O(\log_2 n)$

Ternary → Time Complex

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

$O(\log_3 n)$

$T(n) = T(n/2) + O(1)$

$T(n/2) = T(n/4) + O(1)$

$\vdots$

$T(2) = T(1) + O(1)$

$K = \log_2 n$

$T(n) = T(1) + K\,O(1)$

$T(1) + \log_2 n$

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

$T(\frac{n}{3}) = T(n/9) + O(2)$

$\vdots$

$T(3) = T(1) + O(2)$

$K' = \log_3 n$

$T(n) = T(1) + K'\,O(2)$

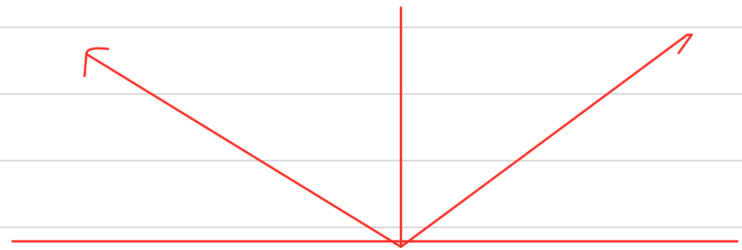
$T(1) + 2\log_3 n$

$\log_2 n \leq 2\log_3 n$

If binary search is better than ternary search is there

any application of ternary search

$f(x) = |x|$



unimodal func^n

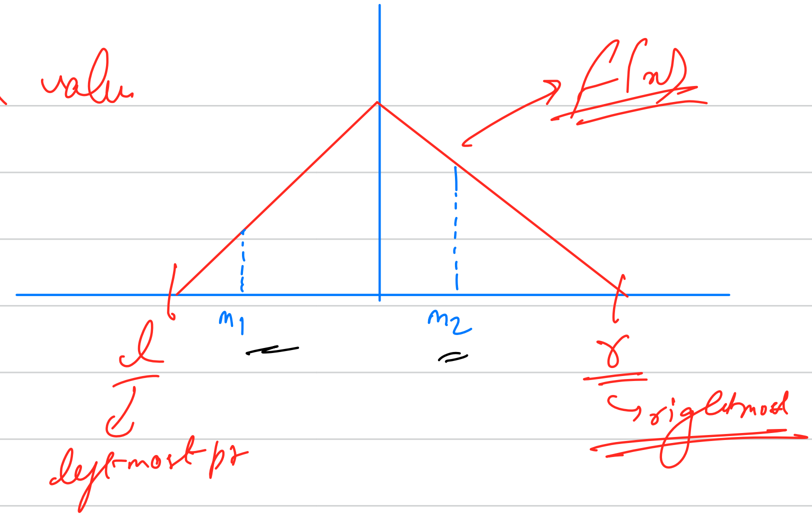
There are 2 type of unimodal f^n

1) The f^n strictly decreases first, reaches a minimum & then strictly increases

2) The f^n strictly increases first, reaches a max & then strictly decrease.

We can use ternary search to find max in case II or min value in case I between [l, r] find the max value of func^n

Let's say $f(x)$ is the func^c



Case I

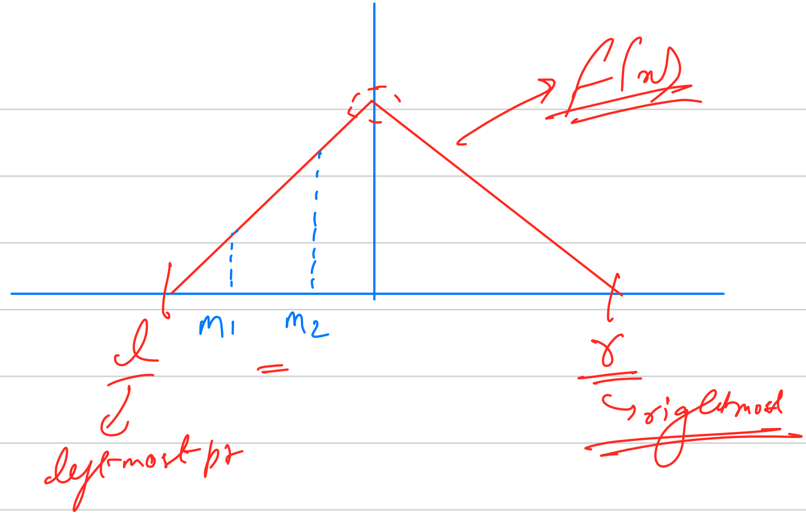
$f(m_1) < f(m_2)$

↳ maxima lies in the ray $[m_1, r]$

Case II

$$f(m_1) < f(m_2)$$

max will lie in the rang $[m_1, r]$

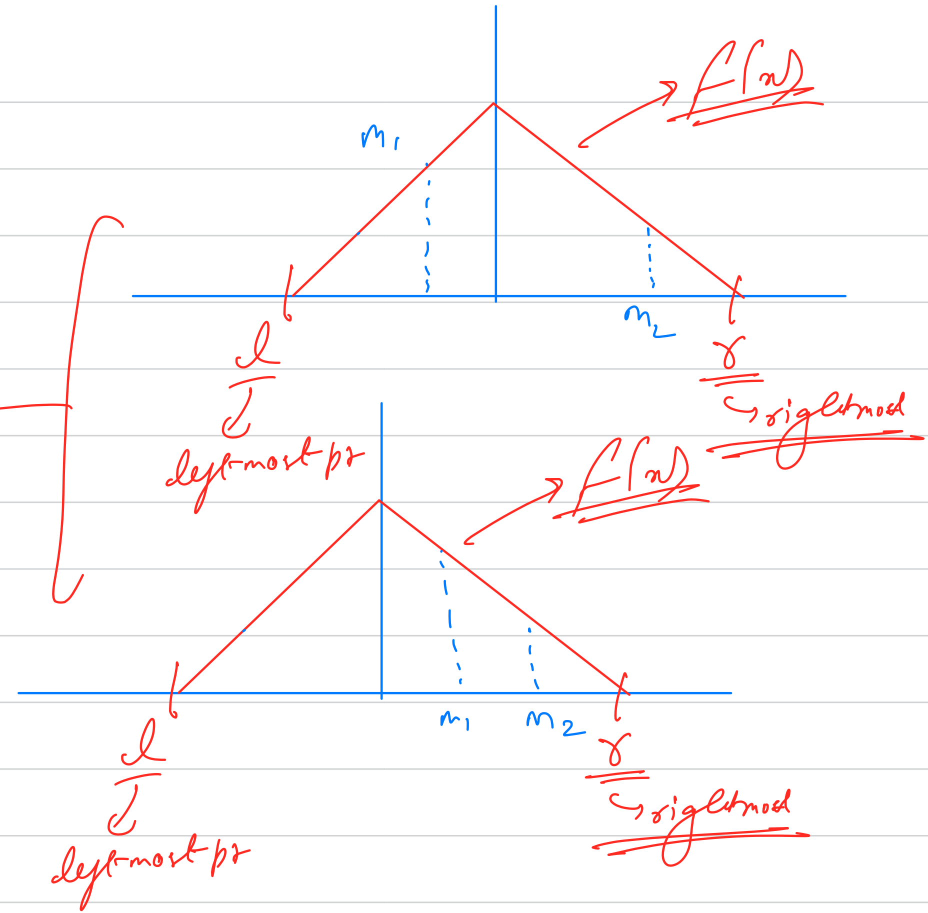


Case III

$f(m_1) > f(m_2)$

max will lie, in the region

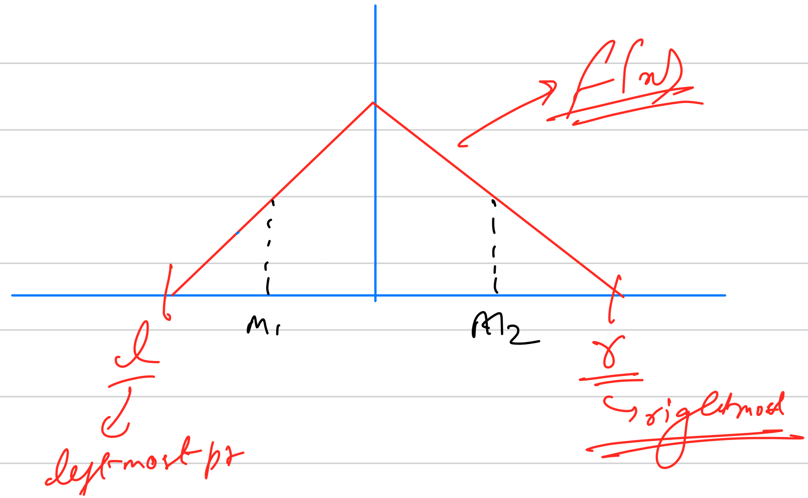
$[l, m_2]$

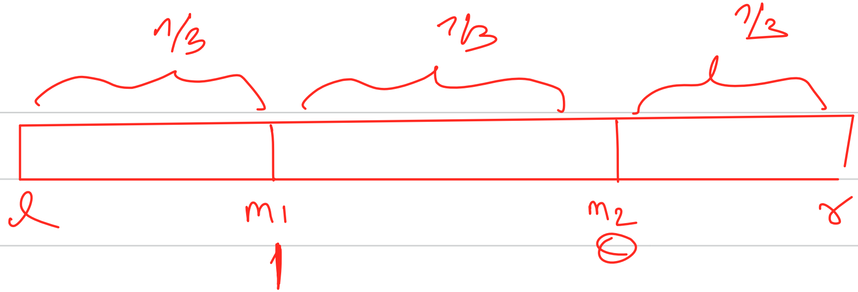


Case IV

$$f(m_1) = f(m_2)$$

max will lie in range $[m_1, m_2]$





$n = \underline{\underline{r - l}}$

$l + \frac{n}{3}$

$m_1 \hookrightarrow l + \frac{(r - l)}{3}$

$m_2 = r - \frac{n}{3}$

$= r - \left(\frac{r - l}{3}\right)$

$$mid = \left(\frac{l+r}{2}\right)$$

$$\Rightarrow \quad l + \frac{r-l}{2}$$

$$l + n/2$$

$$n = r - l$$

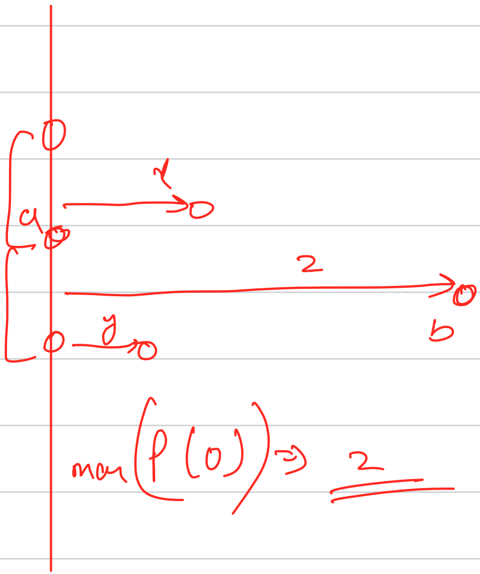
$$\boxed{l + \frac{r-l}{2}}$$

$$f(T) = \underbrace{\max(P_i(T))} - \underbrace{\min(P_j(T))}$$

$$P_i(T) = S_i \times T + O_i$$

at $T=0$

$O_i = 0$

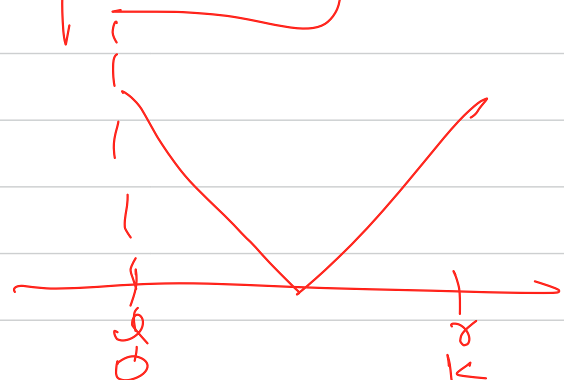


if $S_a > S_b$

$\max(P(0)) \Rightarrow 2$

$\alpha = 0$

$\gamma = K$



will $f(m_1)$ & $f(m_2)$ be always an integer ?

while ($l <= r$)

$l = m_1 + 1$

$r = m_2 - 1$

}

floating point

3.2 4.7

infinite → converge ✗

binary & ternary search on real number

l mid r

$r - l > 10^{-6}$

explore

set no. of iterations

$n = 10^9$

$(\log_2 n) \Rightarrow 30 \to 40$ end

100 iteration $\longrightarrow$ cost