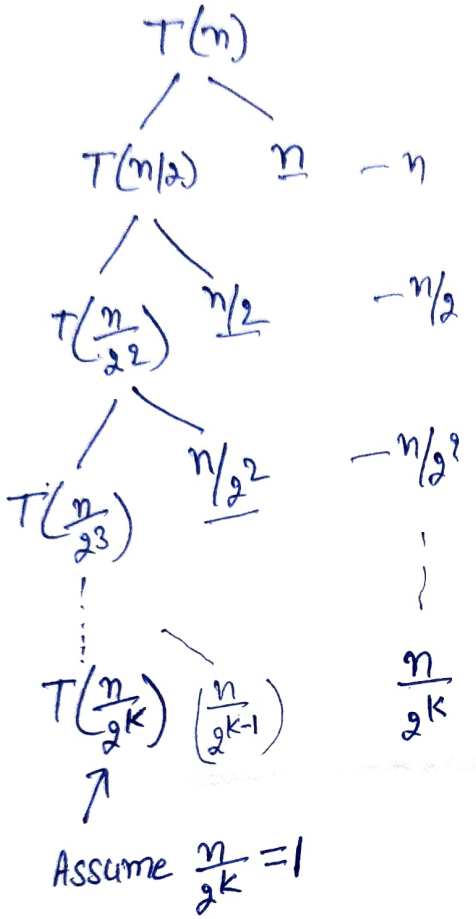
Recurrence Relation (Dividing function) →

Q

$$T(n) = \begin{cases} 1 & n=1 \\ T(n/2)+n & n>1 \end{cases}$$



$$T(n) = n + \frac{n}{2} + \frac{n}{2^2} + \frac{n}{2^3} + \dots + \frac{n}{2^k}$$

$$T(n) = \left[1 + \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \dots \frac{1}{2^k}\right] n$$

$$= n \left(\sum_{i=0}^{k} \frac{1}{2^i}\right) = 1$$

$$= n \times 1$$

$$T(n) = n$$

$$T(n) = T(n/2) + n \quad \text{—①}$$

$$T(n) = \left[T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right] + n$$

$$T(n) = T\left(\frac{n}{2^2}\right) + \frac{n}{2} + n \quad \text{—②}$$

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

$$T(n/2^k) + \frac{n}{2^{k-1}} + \frac{n}{2^{k-2}} + \dots + \frac{n}{2^1} + \frac{n}{2^0}$$

$= 1$ Assume $\frac{n}{2^k} = 1$

$\therefore n = 2^k$ and $k = \log n$

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

$$1 + n[1+1]$$

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

$$= O(n)$$

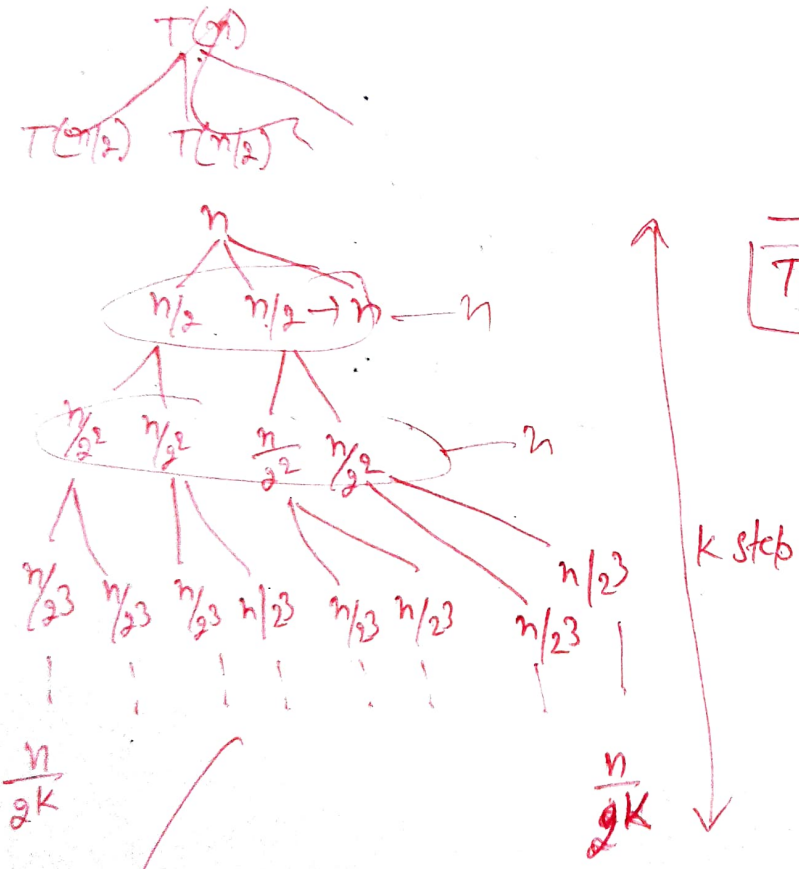
Q Imp

$$T(n) = \begin{cases} 1 & \text{if } n=1 \\ 2T(n/2)+n & \text{if } n>1 \end{cases}$$

```
T(n) — Void Test (int n)
       {  if (n>1)
          {
          for (i=0; i<n; i++)
          {
n —       {  stmt
          }
T(n/2)    Test (n/2)
T(n/2)    Test (n/2)
       }
```

$$\boxed{T(n) = 2T(n/2) + n}$$

Recursion Tree



Assume that $\frac{n}{2^k} = 1$

$n = 2^k$

$k = \log n$

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

Substitution Method.

$T(n) = 2T(n/2) + n$ — ① $T(n/2) = 2T(\frac{n}{2^2}) + n/2$

$T(n) = 2[2T(n/2^2) + \frac{n}{2}] + n$ $T(\frac{n}{2^2}) = 2T(\frac{n}{2^3}) + \frac{n}{2^2}$

$T(n) = 2^2 T(n/2^2) + n + n$ — ②

$T(n) = 2^2[2T(\frac{n}{2^3}) + \frac{n}{2^2}] + 2n$

$= 2^3 T(\frac{n}{2^3}) + n + 2n$

$T(n) = 2^3 T(\frac{n}{2^3}) + 3n$ — ③

$T(n) = 2^k T(\frac{n}{2^k}) + kn.$

Assume $T(\frac{n}{2^k}) = T(1)$

$\therefore \frac{n}{2^k} = 1 \quad n = 2^k$

$k = \log n$

$T(n) = n \times 1 + n \log n$

$O(n \log n)$ Ans

Master theorm for Decreasing function →

example

General form of Recurrence Relation

$$T(n) = a\,T(n-b) + f(n)$$

$a > 0$, $b > 0$ and $f(n) = O(n^k)$ where $k \geq 0$

$T(n) = T(n-1) + 1 = O(n)$ ×n

$T(n) = T(n-1) + n \quad O(n^2)$

$T(n) = T(n-1) + \log n \quad O(n \log n)$

$T(n) = 2T(n-1) + 1 \quad O(2^n)$

$T(n) = 3T(n-1) + 1 \quad O(3^n)$

$T(n) = 2T(n-1) + n \quad O(n 2^n)$

Case 3

If $a = 1$ $O(n^{k+1})$

or $O(n * f(n))$

Case 2

If $a > 1$ $O(n^k a^{n/b})$

If $a < 1$ $O(n^k)$

Case 1 $O(f(n))$

Master Theorm for ~~Recu~~ Dividing function →

$$T(n) = a\,T(n/b) + f(n)$$

$a \geq 1$, $f(n) = \Theta(n^k \log^p n)$

$b > 1$,

① $\log_b a$

② k

Case 1 If $\log_b a > k$ then $\Theta(n^{\log_b a})$

Case 2 If $\log_b a = k$

If $p > -1$ then $\Theta(n^k \log^{p+1} n)$

If $p = -1$ $\Theta(n^k \log \log n)$

If $p < -1$ $\Theta(n^k)$

Case 3 If $\log_b a < k$

a) If $p \geq 0$ $\Theta(n^k \log^p n)$

b) If $p < 0$ $\Theta(n^k$

example case 1 (Master theorm) If $\log_b a > k$ then $\Theta(n^{\log_b a})$

e.g. ① $T(n) = 2T(n/2) + 1$

$a=2 \quad b=2 \quad f(n) = \Theta(1) = \Theta(n^0 \log^0 n)$

$k=0, p=0$

$\log_2 2 = 1 \quad k=0$

So Case 1

$\Theta(n^1)$ Ans

② $T(n) = 4T(\frac{n}{2}) + n$

$\log_2 4 = 2, \quad k=1 \quad p=0$

$\log_2 4 > k$

So case 1

$\Theta(n^2)$ Ans

③ $T(n) = 8T(n/2) + n$.

$\log_2 8 = 3 > k=1$

$\Theta(n^3)$ Ans

④ $T(n) = 8T(\frac{n}{2}) + n^2$

$\log_2 8 = 3 > k=2 \quad p=0$

$\Theta(n^3)$ Ans

⑤ $T(n) = 9T(n/3) + 1$

$\log_3 9 = 2 > k=0$

$\Theta(n^2)$

⑥ $T(n) = 9T(n/3) + n$

$\log_3 9 = 2 > k=1$

$O(n^2)$

⑦ $T(n) = 9T(n/3) + n^2$

$\log_3 9 = 2 = k=2$

Case 2

Example case 2 Master Theorm —

case 2: If $\log_b a = k$

If $P > -1$ $\Theta(n^k \log^{P+1} n)$

If $P = -1$ $\Theta(n^k \log \log n)$

If $P < -1$ $\Theta(n^k)$

e.g (1)

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

$\log_2 2 = 1$ $k = 1$ $P = 0$

$\Theta(n \log n)$

(2) $T(n) = 4T(n/2) + n^2 \log^2 n$

$\log_2 4 = 2$ $k = 2$ $P = 2$

$\Theta(n^2 \log^2 n)$

(3) $T(n) = 8T(n/2) + n^3$

$\log_2 8 = 3$ $k = 3$ $P = 0$

$\Theta(n^3 \log n)$

(4) $T(n) = 2T(n/2) + \frac{n}{\log n}$

$\log_2 2 = 1$ $k = 1$ $P = -1$

$\Theta(n \log \log n)$

example case 3 Master theorm —

① $T(n) = T(n/2) + n^2$

$\log_b a = \log_2 1 = 0 < K = 2$

$\Theta(n^2)$

② $T(n) = T(n/2) + n^2 \log n$

$\log_2 1 = 0 < K = 2 \quad P = 1$

$\Theta(n^2 \log n)$

③ $T(n) =$

Master theorm Examples

① $T(n) = T(\sqrt{n}) + c$

$a=1$ $b=1$ But according to Master theorm b should be ≥ 2. Master theorm will not work if ① $a<1$ ② $b<2$

Master theorm hold for divide and conquer only.

$\therefore$ Assume

$$n = 2^k \Rightarrow k = \log_2 n$$

So $T(n) = T(n^{1/2}) + c$

$$T(2^k) = T(2^{k/2}) + c$$

Assume $T(2^k) = S(k)$

we have $T(2^k) = T(2^{k/2}) + c$

$$S(k) = S(k/2) + c$$

Now we can apply Master Theorm.

$a=1$ $b=2$ $F(n) = c$ i.e $k=0, p=0$

$\log_2 1 = 0$

$k^0 = 1$

$\therefore$ $S(k) = \Theta(\log k)$

If P7-1 then $\Theta(n^k \log^{p+1} n)$

$$T(2^k) = \Theta(\log k)$$

$$T(2^{\log_2 n}) = \Theta(\log \log_2 n)$$

$$T(n) = \Theta(\log \log n).$$ Ans

(2) $T(n) = 2T(\sqrt{n}) + \log n.$

Sol^n Assume $n = 2^k \Rightarrow k = \log_2 n.$

$\therefore T(n) = 2T(n^{1/2}) + \log n$

$T(2^k) = 2T(2^{k/2}) + \log 2^k$

$T(2^k) = 2T(2^{k/2}) + k$

Now Assume $T(2^k) = S(k)$

we have $T(2^k) = 2T(2^{k/2}) + k$

$S(k) = 2S(k/2) + k$

Here $a = 2$
$b = 2$

$\log_b a = \log_2 2 = 1$

$\therefore k^{\log_b a} = k^1 = k$

$S(k) = \Theta(k \log k)$

$T(2^k) = \Theta(k \log k)$

$T(2^{\log_2 n}) = \Theta(\log_2 n \log \log n)$

$T(n) = \Theta(\log \log \log n)$ Ans

Recurrence Relation for Root Function →

$$T(n) = \begin{cases} 1 & \text{if } n=2 \\ T(\sqrt{n})+1 & n>2 \end{cases}$$

```
Void Test(int n)
{
    if (n>2)
    {
        stmt;
        Test(√n)
    }
}
```

$T(n) = T(\sqrt{n}) + 1$

Sol^n

$T(n) = T(\sqrt{n}) + 1$

$T(n) = T(n^{1/2}) + 1$ — (1)

$T(n) = T(n^{\frac{1}{2^2}}) + 2$ — (2)

$T(n) = T(n^{1/2^3}) + 3$ — (3)

⋮

$\cancel{T(n) = T(n^{\frac{1}{2^k}})}$

$T(n) = T(n^{1/2^k}) + k$ — (4)

Assume n is in power of 2

i.e $n = 2^m$

$T(2^m) = T(2^{\frac{m}{2^k}}) + k$

Assume $T(2^{\frac{m}{2^k}}) = 1$ (2^1)

$\therefore \frac{m}{2^k} = 1$

$m = 2^k$ and $k = \log_2 m$

$\because n = 2^m \Rightarrow m = \log_2 n$

$k = \log \log_2 n$

$\Theta(\log \log_2 n)$ Ans

Binary Search →

3	6	8	12	14	17	25	29	31	36	42	47	53	55	62
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Key = 42

l	h	mid $= \lfloor \frac{l+h}{2} \rfloor$
1	15	$\frac{1+15}{2} = 8$
9	15	$\frac{9+15}{2} = 12$
9	11	$\frac{9+11}{2} = 10$
11	11	$\frac{11+11}{2} = 11$

```
int BinSearch(A, n, key)
{   l=1, h=n

    while( l≤h )
    {   mid=(l+h)/2
        IF(key)=A[mid]
            return mid;
        If(key < mid)
            h=mid-1
        else  l=mid+1
    }
    return 0   // not found
```

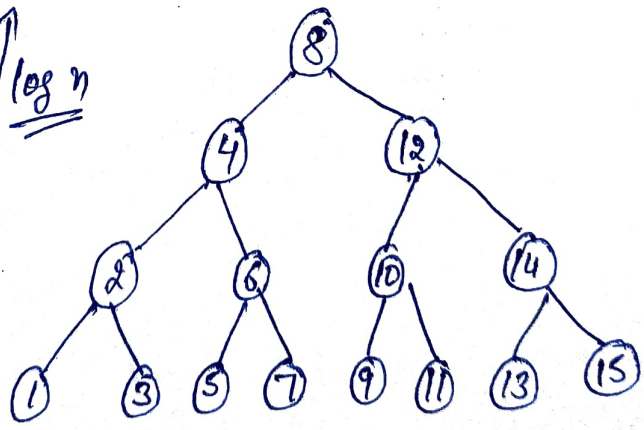
Key = 12

l	h	mid
1	15	$\frac{1+15}{2} = 8$
1	7	$\frac{1+7}{2} = 4$

Key = 30

l	h	mid
1	15	$\frac{1+15}{2} = 8$
9	15	$\frac{9+15}{2} = 12$
9	11	$\frac{9+11}{2} = 10$
9	9	$\frac{9+9}{2} = 9$
9	8	✗ $l > h$.

log n



Max = log n.

Min O(1)

Recursive Binary Search →

$$T(n) = \begin{cases} 1 & n = 1 \\ T(n/2) + 1 & n > 1 \end{cases}$$

Algo RecursiveBinSearch(l, h, key)

```
{
    if (l==h)
    {
        if (A[l]=key)
            return l
        else
            return 0,
    }
    else
    {
        mid = (l+h)/2
        if (key==A[mid]
            return mid,
        if (key<A[mid])
            return (RecursiveBinSearch(l, mid-1, key))
        else
            return(RecursiveBinSearch(mid+1, h, key))
    }
}
```

1

1

1

1