Recurrence Relation

Binary search algo

10, 20, 30, 40, 50, 60, 70
$i = 1$, $j = 7$

BS[a, i, j, x]
mid = (i + j)/2
if (a[mid] == x) ← constant time
  return mid;
else if (a[mid] > x)
  BS[a, i, mid-1, x]
else
  BS[a, mid+1, j, x]

n
n/2 n/2
n/4 n/4
n/8 n/8

So, Recurrence Relation
$T(n) = T(n/2) + c$

Substitution method

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

$T(n) = T(n/2) + c$ — ①
$T(n/2) = T(n/4) + c$ — ②
$T(n/4) = T(n/8) + c$ — ③

$T(n) = T(n/4) + c + c$
$= T(n/2^2) + 2c$

$T(n) = T(n/8) + c + 2c$
$= T(n/2^3) + 3c$
$= T(n/2^4) + 4c$
$= T(n/2^5) + 5c$
k step
$\Rightarrow T(n/2^k) + kc$
$\Rightarrow T(n/n) + kc$
$\Rightarrow T(1) + kc$
$\Rightarrow T(1) + \log n \cdot c$
$\Rightarrow O(\log n)$

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

② $T(n) = \begin{cases} 1 & \text{if } n=1 \\ n*T[n-1] & \text{if } n>1 \end{cases}$

$T(n) = n*T[n-1] \leftarrow ①$

$T(n-1) = (n-1)*T(n-1-1)$

$\Rightarrow (n-1)*T(n-2) \leftarrow ②$

$\Rightarrow$

$T(n-2) \Rightarrow (n-2)*T(n-2-1)$

$\Rightarrow (n-2)*T(n-3) \leftarrow ③$

$T(n) = n*(n-1)*T(n-2)$

$\Rightarrow n*(n-1)*(n-2)*T(n-3)$

$\Rightarrow n*(n-1)*(n-2)*(n-3)*T(n-4)$

$(n-1)$

$T(n-(n-1))$

$\Rightarrow$

$\Rightarrow n*(n-1)*(n-2)*(n-3) \ldots T(1) * 1$

$\Rightarrow n*(n-1)*(n-2)*(n-3)*\ldots\ldots*3*2*1$

$\Rightarrow n*n\left(1-\frac{1}{n}\right)*n\left(1-\frac{2}{n}\right)*n\left(1-\frac{3}{n}\right)\ldots n\left(\frac{3}{n}\right)*n\left(\frac{2}{n}\right)*n\left(\frac{1}{n}\right)$

$\Rightarrow n^n$

$\Rightarrow O(n^n)$

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

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

$T(n/4) = 2T(n/8)+n/4$

$T(n) \Rightarrow 2[2T(n/4)+n/2]+n$

$\Rightarrow 2^2 T(n/2^2)+n+n$

$\Rightarrow 2^2 T(n/2^2)+2n$

$\Rightarrow 2^2[2T(n/8)+n/4]+2n$

$\Rightarrow 2^3 T(n/2^3)+3n$

$\Rightarrow 2^4 T(n/2^4)+4n$

$\Rightarrow 2^5 T(n/2^5)+5n$

$\Rightarrow 2^k T(n/2^k)+kn$

$T(1)$

$n/2^k = 1$

$n = 2^k$

$\log n = \log 2^k$

$\log n = k \log 2$

$\log n = k \cdot 1$

$\Rightarrow 2^k T(1)+kn$

$\Rightarrow n\cdot 1 + n\log n$

$\Rightarrow O(n \log n)$

Master Theorem

$T(n) = aT(n/b) + f(n)$

$a \geq 1,\ b > 1$

$T(n) = n^{\log_b a}\,[u(n)]$

$\Rightarrow u(n)$ depends $h(n)$

$h(n) = \dfrac{f(n)}{n^{\log_b a}}$

$h(n)$	$u(n)$
$n^r,\ r > 0$	$O(n^r)$
$n^r,\ r < 0$	$O(1)$
$(\log n)^i,\ i \geq 0$	$\dfrac{(\log_2 n)^{i+1}}{i+1}$

$\Rightarrow T(n) = 8T(n/2) + n^2$

$a = 8,\ b = 2,\ f(n) = n^2$

$\Rightarrow n^{\log_b a}\ u(n)$

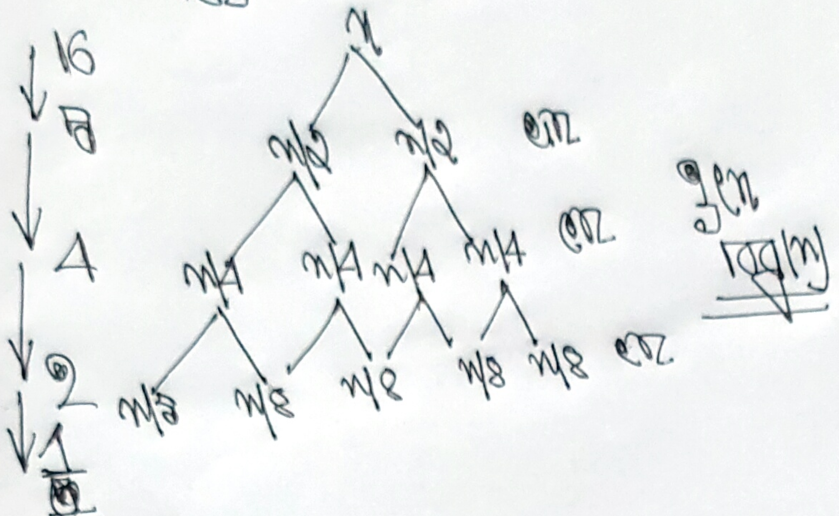
$\Rightarrow n^{\log_2 8}\ u(n)$

$n^3 \cdot 1 = n^3$

$\Rightarrow h(n) = \dfrac{n^2}{n^{\log_2 8}} = \dfrac{n^2}{n^3} = \dfrac{1}{n} = n^{-1}$

Recursive Tree Method

$T(n) = 2T(n/2) + cn$



$\log_2 16 = 4$