2+4+6

DAA (Algo)

Date:

Day: Tuesday

$$for\,(i=1;\ i<=\sqrt{N};\ i *= 2)$$

$\to \log(n)$ $\to \log_2 N$

$$for\,(j=1;\ j<=i;\ j++)\{$$

$$\}$$

$\log^2(n)$

$i=1 \to 1$
$i=2 \to 2$
$i=4 \to 4$
$\vdots$
$i=\sqrt{N} \to \sqrt{N}$

$R=2$
$a=1$
$N=\log_2\sqrt{N}$

→ Arithmetic Series:

$n \to 1+2+3+\cdots+(n-2)+(n-1)+n$

$n/2 \to (n+1)+(n+1)+(n+1)+\cdots+(n+1)$

$$\text{no. of terms} \times (N+1)$$

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

$$= n^2$$

$$= O(N^2)$$

$O(1)$
$\log_2 N$
$\sqrt{n}$
N
$N\log N$
N^2
N^3
$\vdots$
2^N

$$1+2+3+\cdots+\sqrt{n}$$

$$= \frac{\sqrt{n}}{2} \times (\sqrt{n}+1)$$

$$= \frac{N}{2}+\frac{\sqrt{N}}{2} = O(N)$$

$$for\,(i=1;\ i<=N;\ i+k)$$

$\to N/k$

$$for\,(i=1;\ i<=N);\ i *= 2)$$

$\to \log_2 N$

$$for\,(i=1;\ i<=100000;\ i++)$$

$\to O(1)$

Date: Day:

→ Geometric Series:

$$2^0, 2^1, 2^2, \ldots \rightarrow 2^{\log_2 N}$$

$$1, 2, 4, 8, \ldots, N$$

$$2^{\log_2 N} = N^{\log_2 2} \quad \log_2 2 =$$

$$= N$$

$$1 + 2 + 4 + \ldots + N$$

$$\frac{a(R^N - 1)}{R - 1}$$

$\because a = 1$ (1st term)
$\because R = 2$ (Ratio)
$\because N = \log_2 N$ term

$$\frac{1(2^{\log_2 N} - 1)}{2 - 1}$$

$$= 2^{\log_2 N} - 1$$

$$= N^{\log_2 2} - 1 = N$$

$$\log_2 2 = 1$$
$$\log_2 4 = 2$$
$$\log_2 8 = 3$$

$R = 2 : n$
$R = 3 : n^2$
$R = 4 : n^3$

$4^{\log_2 N}$
$N^{\log_2 4}$
$N^{\log_2 2^2}$
$\underline{\underline{N^2}}$

$$1 + 2 + 4 + \ldots + \sqrt{N}$$

$$\frac{1(2^{\log_2 \sqrt{N}} - 1)}{1}$$

$$= 2^{\log_2 \sqrt{N}} - 1 = \sqrt{N}^{\log_2 2} =$$

$$= \sqrt{N}$$

Lecture #02

Date: Aug 21, 25 Day: Thursday

for(int i = 1 ; i <= N ; i += k)

no. of terms generated : $\frac{N}{k}$

if $k = \log_2 N$ $= \frac{N}{\log_2 N}$

for(i = 1 ; i <= N*N ; i += k) $k = \sqrt{N}$

$= \frac{N^2}{k} = \frac{N^2}{\sqrt{N}} = N^{3/2} = N\sqrt{N}$

Arithmatic Series:

for(int i = 1 ; i <= N ; i++)
  for(int j = 0 ; j < i ; j++)
  → $O(N^2)$.

for(i = 1 ; i <= N ; i += 2){
  for(j = 1 ; j <= i ; j++){

i = 1 = 1
i = 3 = 3
i = 5 = 5

no. of terms generated = N/2

→ 1, 3, 5, --- ,(N-4), (N-2) N/2

→ (N+1) + (N+1) + --- (N+1) N/4

$= \frac{N/4(N+1)}{2} = O(N^2)$

Date: Day:

for (i = 1 ; i <= $N^{1/2}$; i += 1)
    for (j = 1 ; j <= i ; j++)

$\sqrt{N}$ $\quad 1+2+3+ --- + (\sqrt{N}-1) + \sqrt{N}$

$\sqrt{N}/2$ $\quad (\sqrt{N}+1) + (\sqrt{N}+1) + --- (\sqrt{N}+1)$

$$= \frac{\frac{\sqrt{N}}{2} \times (\sqrt{N}+1)}{2}$$

$$= O(N)$$

for (i = 1 ; i <= N ; i *= 2)
    for (j = 1 ; j <= i ; j *= 2)

1, 2, 4, 8, 16, ----- N

i = 1 : 1, — 1
i = 2 : 1, 2 — 2
i = 4 : 1, 2, 4 — 3
i = 8 : 1, 2, 4, 8 — 4

* $1+2+3+4+ --- + \log N$

no. of terms = $\log n$ $\quad r = 2$

1st term = 1

$$= \frac{a(r^n - 1)}{r-1} \qquad \frac{1(2^{\log N} - 1)}{2-1}$$

$$= 2^{\log_2 N} = N^{\log_2 2} = N$$

Date: Day:

$$1+2+3+ \cdots + \log N = (\log_2 N)^2$$

$$= \frac{\log N(\log N + 1)}{2} = (\log_2 N)^2$$

```
for(i = 1; i <= log2N; i++)
    for(j = 1; j <= i; j++)
```

$$\rightarrow (\log_2 N)^2$$

Arithmatic : $N^k * N^k$

```
for(i = 1; i < N; i++)
    for(j = 1; j <= i^2; j++)
```

$$\rightarrow 1^2 + 2^2 + 3^2 + \cdots + N^{2} = O(N^3) \quad k=2$$

$$= \frac{N*(N+1)(2N+1)}{6} = \boxed{N^{k+1}} \quad k=2$$

$1^k + 2^k + 3^k + \cdots N^k : N^{k+1}$

$1, 2, 4, 8, \cdots, N$

$2^0, 2^1, 2^2, 2^3, \cdots 2^k$

$2^k = N$

$\log_2(2^k) = \log_2 N$

$k \times \log_2 2 = \log N$

$\boxed{k = \log_2 N}$

Date: Day:

Geometric Series sum:

$$= \frac{a(r^n - 1)}{r - 1}$$

for(i = 1 ; i <= N^2 ; i *= 2)
    for(j = 1 ; j <= i ; j++)

no. of terms = $\log_2 N^2$

$$= \frac{1 \times (2^{\log_2 N^2} - 1)}{2 - 1}$$

$$= \frac{N^{2\log_2 2} - 1}{1} = \boxed{N^2}$$

Overestimation:

for(i = 1; i <= N; i++
    for(j = 1 ; j <= i ; j++)
        for (k = 1 ; k <= j ; k++)

i = 1	j = 1	k = 1	1	1
i = 2	j = 2	k = 2	3	(1+2)
⋮	⋮	⋮	6	(1+2+3)
			⋮	
i = N	j = N	k = N		(1+2+3+ -- N)

$$\underbrace{\qquad\qquad}_{(N^2)}$$

$\cancel{N(N^2)} = N^3$

$$1 + (1+2) + (1+2+3) + \cdots + (1+2+3+ \cdots N)$$

Date: Day:

$N^2 + N^2 + \cdots + N^2$

$= N(N^2) = N^3$

```
for ( i = 1; i < n; i *= 5)
    for ( j = 1; j <= i; j + 2)
```

$\left[\frac{i}{2}\right]$

$i = 1$	~~$j = 0, 1$~~ $j = 1$	1
$i = 5$	$j = 1, 3, 5$	3
$i = 25$	$j = 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25$	13 $\left[\frac{25}{2}\right]$
$\underbrace{i = N}_{\log_5 N}$	$\underbrace{j = 1, 3, 5, \ldots, N}$	$\left[\frac{N}{2}\right]$

~~$1 + (1+2) + \cdots (1+2+$~~

$= \frac{1}{2} + \frac{5}{2} + \frac{25}{2} + \cdots + \frac{N}{2}$

$= \frac{1}{2}\left(1 + 5 + 5^2 + 5^3 + \cdots N\right)$

$= \frac{a(r^N - 1)}{r - 1}$

$= 1\left(\frac{5^{\log_5 N} - 1}{5 - 1}\right)$

$= 1\left(\frac{N^{\log_5 5} - 1}{4}\right) = N$

```
k = 0;
for( j = 1 ; j*j <= n ; j++)
    k++;
```

Aug 26, 25 Tues

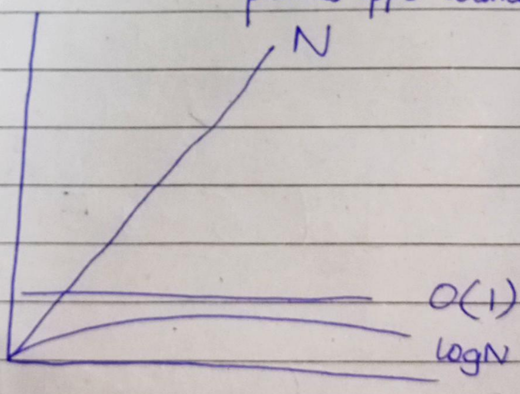
Upper Bound:

$T(N) = 3N^2 + 4N + 5$

$f(N) \leq c * g(N)$ for all $N \geq n_0$.

const, $O(1)$, ~~O(N)~~, $\log(N)$, $\sqrt{N}$, N, $N * \log N$, N^2, N^3, ---- 2^N, 3^N, ----

provide Lower bound. provides upper bound.



for: $f(N) = 3N^2 + 4N + 5$

$O(N^3)$ $\Theta(N^2)$

$\Omega(N)$

Date: Day:

$f(N) \leq c * g(N)$

$f(N) = 3N^2 + 4N + 5$

$g(N) = \{N^3, N^4, \ldots, 2^N, 3^N\}$

$g(N) = \{2^N\}$, $c = 1$

$3N^2 + 4N + 5 \leq 1 * 2^N$

'no': the input value ~~function~~ on which it gives the upper bound (always)

in this case $N = 8$ is no

($N \geq n_0$)

$f(N) \geq c * g(N)$

$c = 40$, $g(N) = N$

$3N^2 + 4N + 5 \geq 40(N)$

in this case it provides Lower bound.

at $N = 12$ the above function provides the Lower bound.

$f(N) \geq c_1 * g(N)$

$f(N) \leq c_2 * g(N)$

agar hm $f(N) == g(N)$ Tight bound...

$f(N) = 3N^2 + 4N + 5$

$g(N) = N^2$

provides both upper & lower bounds which depends on constant 'c'

$c_1 * g(N)$

$\Theta(N^2)$ Tight bound.

Date: $\Omega(N)$ = Lower bound Day:

$$f(N) = \log_2 N$$

$$g(N) = \log_8 N$$

$$f(N) = O(g(N))$$
$$= \Omega(g(N))$$
$$= \Theta(gN) = ?$$

Ans:

$g(N)$ provides ~~Tight Bound~~. TightBound.

$$\log_8 N = \frac{\log_2 N}{\log_2 8}$$

$$\log_8 N = \frac{\log_2 N}{3}$$

$$3 \times \log_8 N = \log_2 N$$

TightBound: Provides lower & Upper both depending on the value of 'C'

if:
$$\left.\begin{aligned} f(N) &= 8^N \\ g(N) &= 2^{3N} \end{aligned}\right] \text{TightBound...}$$

if:
$$\left.\begin{aligned} f(N) &= 2^{N+10} \\ g(N) &= 2^N \end{aligned}\right] \text{Tight Bound}$$

$$f(N) \geq c * g(N)$$
$$2^N \times 2^{10} \geq c * 2^N$$

Date: ________ Day: ________

$$\frac{2^N \times 2^{10}}{2^N} \geq c$$

$2^{10} \geq c$ if we take $c \leq 2^{10}$: Lower bound

if $c \geq 2^{10}$: Upper bound.

$$\left.\begin{aligned} f(N) &= 2^{10N} \\ g(N) &= 2^N \end{aligned}\right]$$ Lower $f(N) = \Omega(g(N))$

$$f(N) \leq c * g(N)$$

$$2^{10N} \leq c * 2^N$$

$$\frac{2^{10N}}{2^N} \leq c$$

$$\boxed{2^{9N} \leq c}$$ 'c' value depends on 'N'

"Recursive Algorithm"

```
f(N) { if  N == 1
            return ;        ] base case
     f(N-1);
     for (i = 1; i <= 100000 ; i++) {
            - - - -
     }
```

T(N) = no. of recursive calls:

$$T(N) = T(N-1) + \Theta(1)$$

Dividing larger problems

Breaking into smaller problems

Time complexity for solving each subproblem

$$
\begin{aligned}
N &- 1 \\
&+ \\
N-1 &- 1 \\
&+ \\
N-2 &- 1 \\
&+ \\
&\vdots \\
3 &- 1 \\
&+ \\
2 &- 1 \\
&+ \\
1 &- 1
\end{aligned}
$$

$= N * 1$ "N : no. of levels"

$= N$

```
f(N) { if N==1
  f(N-1)
  for(i=1; i<=N; i++){
     ---
  }
}
```

$$
\begin{aligned}
N &- N \\
N-1 &- N-1 \\
&\vdots \\
3 &- 3 \\
2 &- 2
\end{aligned}
$$

$N+(N-1)+(N-2)+ \cdots +3+2+1$

$O(N^2)$

Arithmatic Series.

- Recursive calls (no. of)
- Cost of each sub-problem

Date: Day:

~~for (i~~ f(N) { if N==1
return ;

f(N/2)
f(N/3)
f(N-2)

```
for (i=1; i<=N; i++){
   for (j=1; j<=N; j++){
      for (k=1; k<=N; k++)
         ≡

for (x=1; x<=N; x*=2)
   for (y=1; y<=N; y++){

   }
}
```

$$T(N) = T(N/2) + T(N/3) + T(N-2) + \dots N^3$$

$$T(N) = T(N-2) + N$$

N — N

N-2 — N-2

N-4 — N-4

⋮

3 — 3

1 — 1

$1+3+5+\dots+(N-4)+(N-2)+N$

Arithmatic Series with a diff of 2

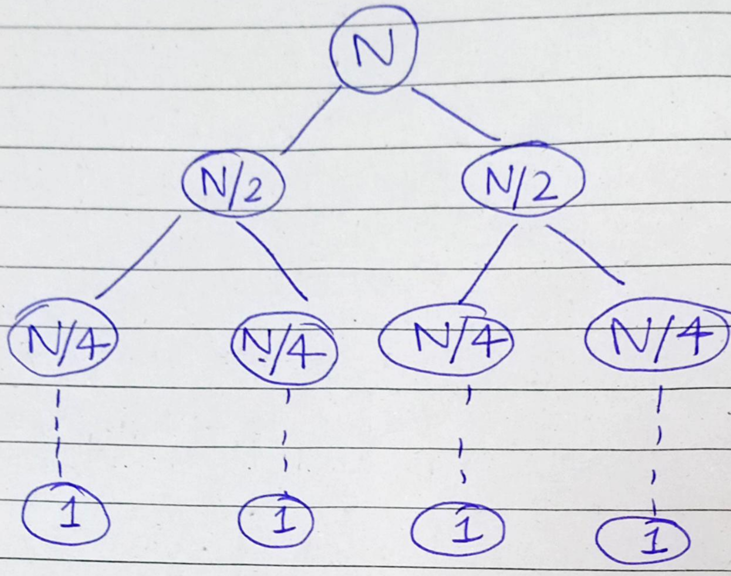
$O(N^2)$

Date: Day:

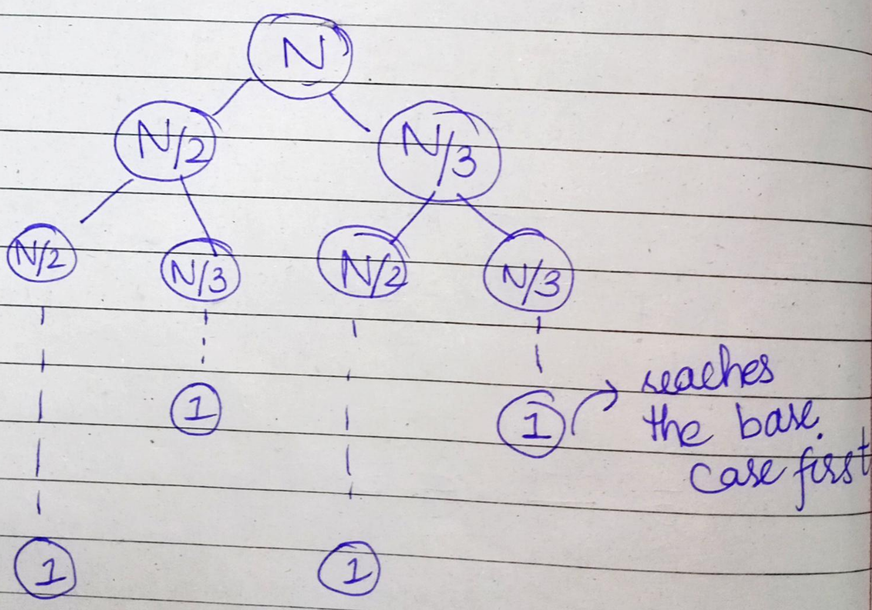
$$T(N) = T(N/2) + T(N/2) + N$$

no. of recursive call

$$T(N) = 2\,\underbrace{T(N/2)}_{\text{input size}}$$

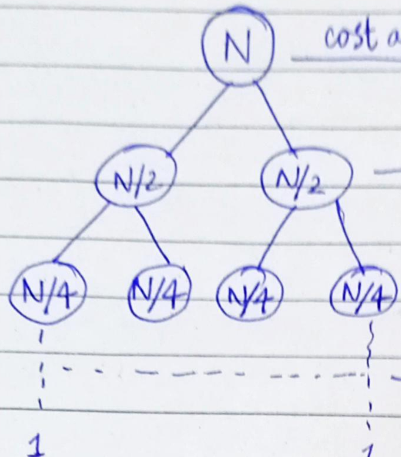


$$T(N) = T(N/2) + T(N/3) + N$$



Recursion Tree for binary Search:

$$T(N) = \cancel{T(s, N-1, ks) + T(N+1, \ldots)}$$
$$= T(N/2) + T(N/2) + O(1).$$



cost at each lvl

1

— 1+1=2

1+1+1+1 = 4

---- 8*1=8

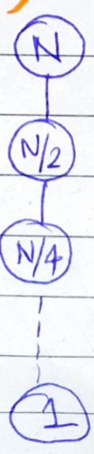
— N*1 = N

$$1+2+4+8+ \cdots + N$$

$R=2$
$a=1$
$N=\log N$

$$= \frac{a(R^N-1)}{R-1} = \frac{1(2^{\log N}-1)}{2-1}$$
$$= 2^{\log_2 N}$$
$$= N \quad \text{(incorrect).}$$

✱ $T(N) = T(N/2) + O(1)$



— 1

— 1

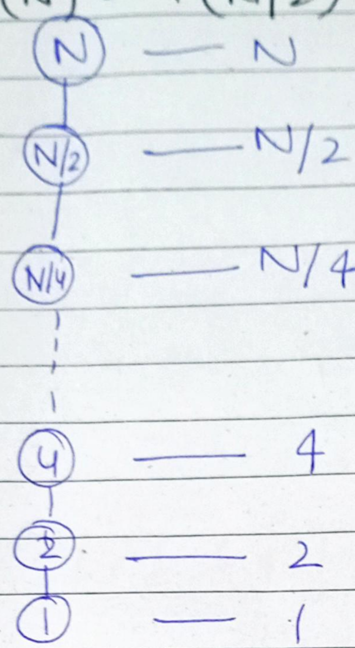
— 1

— 1

$$= O(\log_2 N)$$

Date: Day:

$T(N) = T(N/2) + N$



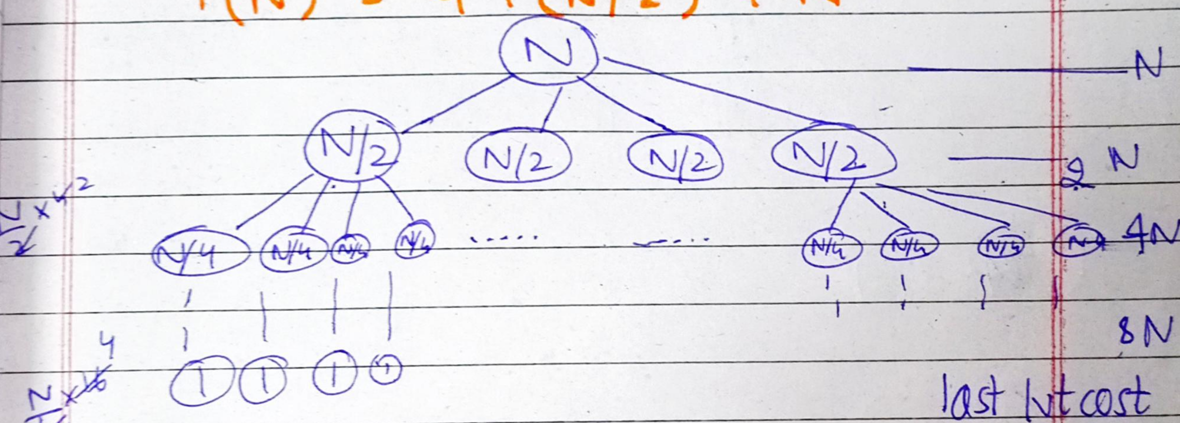
Geometric Series

$= N + \frac{N}{2} + \frac{N}{4} + \dots + 4 + 2 + 1$

$= N$

N
N/4
N/16

$T(N) = 4\,T(N/2) + N$



last lvl cost $= 4^k \times 1$

$4^k \times 1$

$N + 2N + 4N + 8N + \dots \underline{N \times N}$

$= N2$

$= \frac{a(R^N - 1)}{R - 1} = \frac{N(2^{\log_2 N} - 1)}{2 - 1}$

$= N(2^{\log_2 N}) = N(N)$

$= N^2$

$$\frac{N}{2^k} = 1$$

$$N = 2^k$$

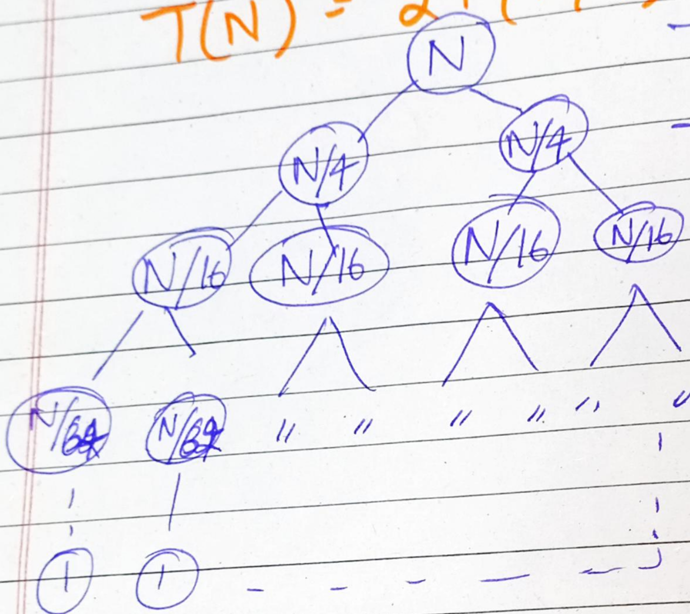
$$\log_2 N = \log_2 2^k$$

$$= k \times \log_2 2$$

$$\log_2 N = k$$

$$\boxed{k = \log_2 N}$$

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



N

$2\left(\frac{N}{4}\right) = \frac{N}{2}$

$4(N/16) = \frac{N}{4}$

$= N/8$

$\frac{N}{2^k}$

$$N + \frac{N}{2} + \frac{N}{4} + \cdots + \frac{N}{2^k}$$

$$= N\Big(\underbrace{1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots \frac{1}{2^k}}_{\hookrightarrow 1/(1-R)}\Big) \quad \because R < 1$$

↳ diression

Date: ________ Day: ________

Geometric Series :

$$\left(\sum_{i=1}^{\infty} \frac{1}{2^i} \right) = N\left(\frac{1}{1-R} \right)$$

$$= N\left(\frac{1}{1-1/2} \right) = O(N)$$