Algo - Lecture 10 - Solving Recurrences

- **Reminder**
  - **Arithmetic Sequence**

    1. 1,2,3,4,...,N — are O(N) terms
    2. 0,2,4,6,...,N — are O(N) terms
    3. 0, k, 2k, 3k, … , N — are O(N/k) terms, if k = constant i.e. O(N) terms
    4. 0, log N, 2log N, 3log N, …., N — are O(N/log N) terms
  - **Arithmetic Series and relatives**

    $1+2+3+4+...+N = O(N^2)$

    $1+2+3+4+...+\log N = \log N \times \log N = O(\log^2 N)$

    $1^2+2^2+3^2+...+N^2 = O(N^3)$

    $1+2+3+4+...+N^2 = O(N^4)$

    $1+2+3+4+...+T = O(T^2)$ where $T = \log N$

    $1^k+2^k+3^k+...+N^k = O(N^{k+1})$
  - **Geometric Sequence**

    1. 1, 2, 4, 8, 16, …., N — are O(log N) terms
    2. 1, 3, 9, 27, …… , N — are O(log N) terms
    3. $1, k, k^2, k^3, ……., N$ — are $O(\log_k N)$ Terms
  - **Geometric Series**

    1. 1+2+4+8+.....+N — <= 2N is O(N)

       N+N/2+N/4+...+8+4+2+1

       =N(1+½+¼+⅛+.........+2/N+1/N)
    2. $1+3+3^2+3^3+...+N$ — is O(N)
    3. $1+3+3^2+3^3+...+N^2$ — $\le 2N^2$ is $O(N^2)$
    4. $1+3+3^2+3^3+...+T$ — <= 2T is O(T)

    Proof

    $$S_n = ar^0+ar+ar^2+ar^3+ar^4+...+ar^{N-1}$$
    $$rS_n = \quad ar+ar^2+ar^3+ar^4+...+ar^{N-1}+ar^N$$
    $$-\ \rule{10em}{0.4pt}$$
    $$S_n-rS_n = a- ar^N$$
    $$S_n(1-r) = a(1-r^N)$$
    $$Sn = \frac{a(1-r^N)}{(1-r)} = \text{ for } r = 2 \Rightarrow 1(1\text{-}2N)/(1\text{-}2) = 2N\text{-}1$$

    $$\text{Lim}_{N\Rightarrow \text{inf}}\ c^N \Rightarrow 0 \quad \text{if } 0<|c|<1$$
    $$Sn = \frac{a}{(1-r)} \quad \text{when } |r|<1$$

int Sum(int N)
{
  if(N==0) return 0;   O(1)
  return Sum(N-1) + N;   O(1)
}
	BubbleSort(A, N)
	{
	  if(BubbleUp(A, N)) O(N)
	    BubbleSort(A, N-1);
	}
		Bool BubbleUp(A, N)
		{
		  Bool CH = false;
		  for(int i=0; i+1<N; i++)
		  {
		    if(A[i]>A[i+1])
		    {Swap(A[i], A[i+1]); CH = true; }
		  }
		  return CH;
		}

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

$T(N-1) = T(N-2) \quad + 1$
$T(N-2) = T(N-3) \quad + 1$
$T(N-3) = T(N-4) \quad + 1$
$T(N-4) = T(N-5) \quad + 1$
.
.
.
$T(2) \quad = T(1) \; + \; 1$
$T(1) \quad = T(0) \quad + 1$

$1+1+1+....+1 \qquad = N$

$T(N) = T(N-1) + O(N)$

$T(N-1) = T(N-2) \quad + N-1$
$T(N-2) = T(N-3) \quad + N-2$
$T(N-3) = T(N-4) \quad + N-3$
$T(N-4) = T(N-5) \quad + N-4$
.
.
.
$T(2) \quad = T(1) \; + \; 2$
$T(1) \quad = T(0) \quad + 1$
$T(0) \quad = \qquad 0$

$1+2+3+4+5+...+N = O(N^2)$

$T(N) = T(N-1) + N^2$

$T(N-1) = T(N-2) \quad + (N-1)^2$
$T(N-2) = T(N-3) \quad + (N-2)^2$
$T(N-3) = T(N-4) \quad + (N-3)^2$
$T(N-4) = T(N-5) \quad + (N-4)^2$
.
.
.
$T(2) \quad = T(1) \; + \; 2^2$
$T(1) \quad = T(0) \quad + 1^2$
$T(0) \quad = \qquad 0$

$1^2+2^2+3^2+...+N^2 = O(N^3)$

$T(N) = T(N/2) + 1$

$T(N/2) = T(N/4) +1$
$T(N/4) = T(N/8) +1$
$T(N/8) = T(N/16) +1$
$T(N/16) = T(N/32) +1$
.
.
.
$T(2) \quad = T(1) \; + 1$
$T(1) = \qquad +1$

$1+1+1+....+1 \qquad = \log N$

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

$T(N/2) = T(N/4) + \; N/2$
$T(N/4) = T(N/8) + N/4$
$T(N/8) = T(N/16) +N/8$
$T(N/16) = T(N/32) +N/16$
.
.
.
$T(2) \quad = T(1) \; + 2$
$T(1) = \qquad + 1$

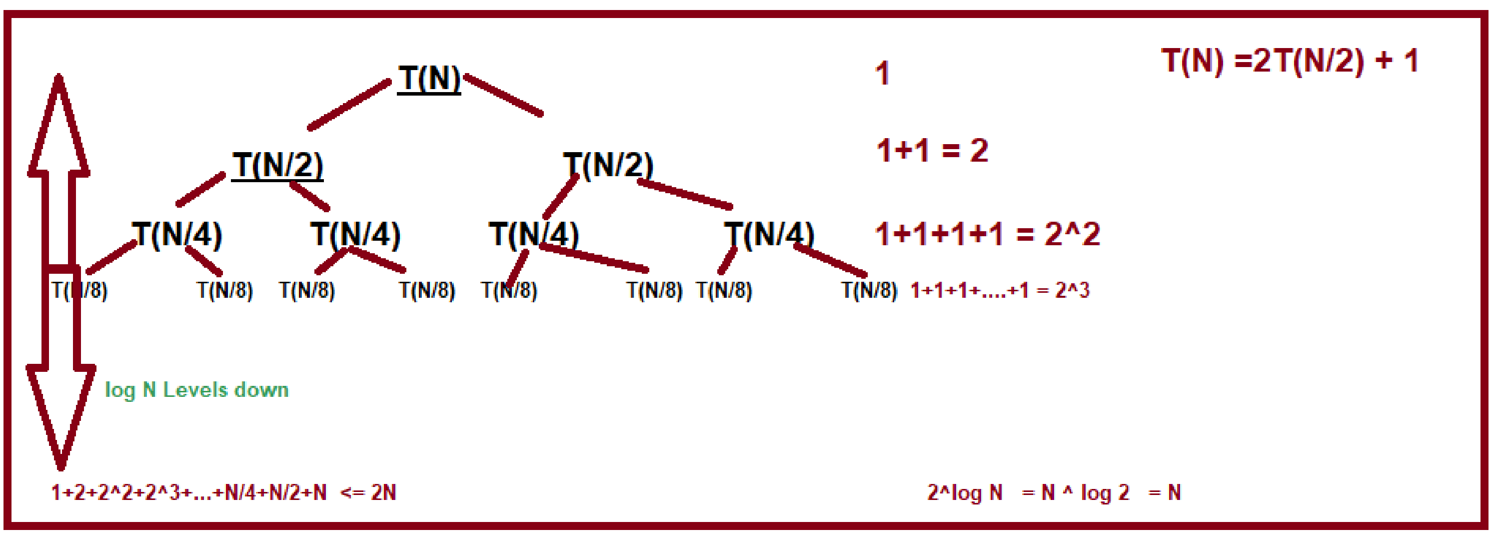
$1+2+4+8+...+N <= 2N$

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

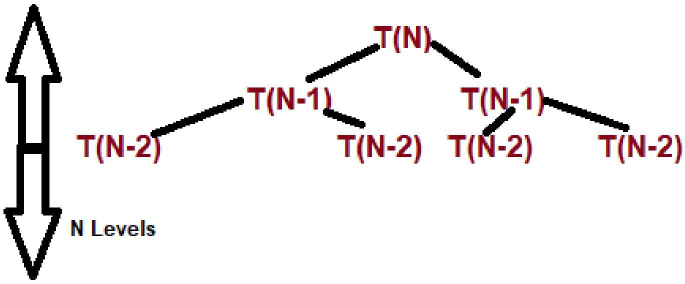
$T(N/2) = T(N/4) + \; (N/2)^2 = N^2/4$
$T(N/4) = T(N/8) + N^2/16$
$T(N/8) = T(N/16) +N^2/64$
$T(N/16) = T(N/32) +N^2/256$
.
.
.
$T(2) \quad = T(1) \; + 2$
$T(1) = \qquad + 1$

$N^2(1+¼+1/4^2 + 1/4^3+1/4^4+....+1/N^2) <= 2N^2 = O(N^2)$

$T(N) = 2T(N/2) + 1$



$T(N) = 2T(N-1) + 1$



1

2

2^2

2^3

T(N) =2T(N-1) + 1

1+2+2^2+....+2^N   <= 2.2^N = 2^(N+1)

Solve the recurrence for T(N) =3T(N-1) + 1 QUIZ

$$T(N) = a\,T(N/b) + N^c$$

Master Theorem

1. If the cost remain constant over each level then the total cost is O(one level cost) x # of levels $(N^c \times \log_b N)$
2. If the cost is decreasing over each level geometrically then the total cost is the cost of 1st step. $O(N^c)$
3. If the cost is increasing over each level geometrically then the total cost is the cost of the last step $(a^{\log_b N})$.

Total Cost on layer 0 $= N^c$

Total Cost on this layer 1 $= a * (N/b)^c$

Total Cost on Layer 2 $= a\text{^}2 . (N/b\text{^}2)^c$

Total Cost on Layer 3 $= a\text{^}3 . (N/b\text{^}3)^c$

Total Cost on Layer 4 $= a\text{^}4 . (N/b\text{^}4)^c$

....

Total Cost on Layer # $\log_b N = a^{(\log_b N)} \times O(1) = N^{(\log_b a)}$

$$= N^c + a * (N/b)^c + a\text{^}2 . (N/b\text{^}2)^c + a\text{^}3 . (N/b\text{^}3)^c + a\text{^}4 . (N/b\text{^}4)^c + \ldots + a^{(\log_b N)} (N/b\text{^}\log_b N = O(1))^c$$

$$= N^c + a (N^c/b^c) + a^2 N^c/b^{2c} + a^3 N^c/b^{3c} + a^4 N^c/b^{4c} + \ldots + a^{(\log_b N)} N^c/b^{(\log_b N).c}$$

$$\boxed{= N^c (1 + a/b^c + (a/b^c)^2 + (a/b^c)^3 + (a/b^c)^4 + \ldots + (a/b^c)^{\log_b N}) \Rightarrow (A)}$$

<u>Case 1 => $a/b^c = 1$</u>

$= N^c (1 + 1 + (1)^2 + (1)^3 + (1)^4 + \ldots + (1)^{\log_b N}) = O(N^c . \log_b N)$

$T(N) = 2T(N/2) + N^1$ $a = 2$ $b=2$ $c = 1$

$2/2^1 = 1$

$O(N \times \log_2 N)$

$T(N) = 4T(N/2) + N^2$ $4/2^2 = 1$

$O(N^2 . \log_2 N)$

$T(N) = 27T(N/3) + N^3$ $27/3^3 = 1$

$O(N^3 . \log_3 N)$

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

$T(N)$ — N

$T(N/2)$ $T(N/2)$ — $N/2 + N/2 = N$

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

Total Cost = N x (log_2 N)

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

$T(N)$ ===> N^2

$T(N/2)$ $T(N/2)$ $T(N/2)$ $T(N/2)$ ==> 4x(N/2)^2 = N^2

Total Cost = N^2 x log_2 N

<u>Case 2 => a/b^c < 1 = 0.5 = 1/2</u>

$= \quad N^c\,(1 + a/b^c + (a/b^c)^2 + (a/b^c)^3 + (a/b^c)^4 + \ldots + (a/b^c)^{\log_b N}) \Rightarrow (A)$

Example $a/b^c = 0.5 = ½$

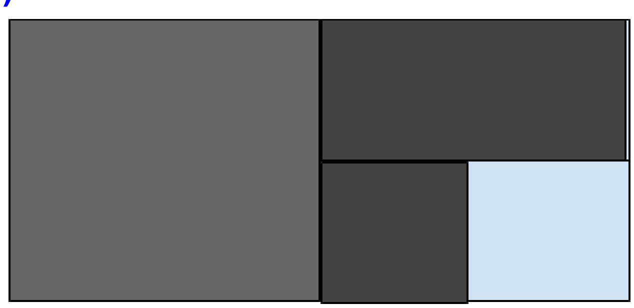
$N^c\,(1 + ½ + (½)^2 + (½)^3 + (½)^4 + \ldots + (½)^{\log_b N}) \Rightarrow (A)$

$N^c + N^c/2 + N^c/4 + N^c/8 + \ldots\ldots\ldots + 8+4+2+1 <= 2N^c = O(N^c)$

Reminder:

$N + N/2 + N/4 + N/8 + N/16 + \ldots\ldots$

$N(1+½ + ¼ + ⅛ + 1/16 + \ldots) = N \,.\, 2 = 2N$



$1+½ + ¼ + ⅛ + 1/16 + \ldots <= 2$

$= N^c/(1-a/b^c) = O(N^c)$

Example

$T(N) = 4T(N/8) + N$

$a/b^c = 4/8^1 = ½ = O(N)$

T(N) = 4T(N/8) + N

T(N) | N

T(N/8) T(N/8) T(N/8) T(N/8) | 4.N/8 = N/2

T(N/64) T(N/64) T(N/64) T(N/64)... T(N/64) T(N/64) T(N/64) T(N/64) | 16xN/64 = N/4

Reminder
N+N/2+N/4+N/8+........................+ 8+4+2+1 <=2N

N+N/2+N/4+N/8+........+ N^(log_8 4) <= 2N

```
for(int i=N;    i>=1 ; i/=2)    O(log N)
        for(int j=1; j<=i; j++)       O(N)

for(int i=N;    i>=sqrt(N) ; i/=2)    O(log N)
        for(int j=1; j<=i; j++)       O(N)
```

Total Cost = O(N)

Case 3 => $a/b^c > 1$ e.g. 1.5

$= N^c (1 + a/b^c + (a/b^c)^2 + (a/b^c)^3 + (a/b^c)^4 + \ldots + (a/b^c)^{\log_b N}) \Rightarrow (A)$

$= N^c (1 + a/b^c + (a/b^c)^2 + (a/b^c)^3 + (a/b^c)^4 + \ldots + (a/b^c)^{\log_b N})$

Reminder:

Geometric Series, 1. if Decreasing, Bounded by the first term

2. if increasing, Bounded by the last term

$= N^c ((a/b^c)^{\log_b N})$

$= N^c (a^{\log_b N} / b^{c \log_b N})$

$= N^c (a^{\log_b N} / N^{c \log_b b}) = \cancel{N^c} (a^{\log_b N} / \cancel{N^c}) = a^{\log_b N}$