Data Structure and Algorithms

Session-17

Dr. Subhra Rani Patra
SCOPE, VIT Chennai

Theorem (Master Theorem)

Let $T(n)$ be a monotonically increasing function that satisfies

$$T(n) = aT(\frac{n}{b}) + f(n)$$

```
where,
n = size of input
a = number of subproblems in the recursion
n/b = size of each subproblem. All subproblems are assumed
     to have the same size.
f(n) = cost of the work done outside the recursive call,
      which includes the cost of dividing the problem and
      cost of merging the solutions

Here, a ≥ 1 and b > 1 are constants, and f(n) is an asymptotically positive function.
```

Recurrence

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

$a \geq 1$, $b > 1$, $k \geq 0$

p = Real number.

1) If $a > b^k$, then $T(n) = \Theta\left(n^{\log_b^a}\right)$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \Theta\left(n^{\log_b^a} \log^{p+1} n\right)$
   b. If $p = -1$, then $T(n) = \Theta\left(n^{\log_b^a} \log\log n\right)$
   c. If $p < -1$, then $T(n) = \Theta\left(n^{\log_b^a}\right)$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \Theta(n^k \log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

You *cannot* use the Master Theorem if

- $T(n)$ is not monotone, ex: $T(n) = \sin n$
- $f(n)$ is not a polynomial, ex: $T(n) = 2T(\frac{n}{2}) + 2^n$
- b cannot be expressed as a constant

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

$a = 4,\ b = 2,\ k = 1,\ p = 0$

$4 > 2^1$

$$T(n) = \Theta(n^{\log_2 4})$$

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

$$T(n) = 16^n T\left(\frac{n}{3}\right) + \frac{1}{n}$$

$a = 16^n \quad b = 3 \quad k = -1$

1) If $a > b^k$, then $T(n) = \Theta\left(n^{\log_b^a}\right)$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \Theta\left(n^{\log_b^a} \log^{p+1} n\right)$
   b. If $p = -1$, then $T(n) = \Theta\left(n^{\log_b^a} \log\log n\right)$
   c. If $p < -1$, then $T(n) = \Theta\left(n^{\log_b^a}\right)$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \Theta(n^k \log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

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

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

$2 = 2^1$

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

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

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

$= 2T\left(\frac{n}{2}\right) + n.\log^{-1} n$

$a = 2, \ b = 2, \ k = 1, p = -1$

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

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

1) If a > b^k, then $T(n) = \Theta\left(n^{\log_b^a}\right)$
2) If a= b^k
   a. If $p > -1$, then $T(n) = \Theta\left(n^{\log_b^a} \log^{p+1} n\right)$
   b. If $p = -1$, then $T(n) = \Theta\left(n^{\log_b^a} \log \log n\right)$
   c. If $p < -1$, then $T(n) = \Theta\left(n^{\log_b^a}\right)$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \Theta(n^k \log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

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

$$= 2T\left(\frac{n}{2}\right) + n\log^{-2} n$$

$a = 2 \quad b = 2 \quad k = 1, \; p = -2$

$\Theta\left(n^{\log_2 2}\right)$

$= \Theta(n)$

1) If a > b^k, then $T(n) = \Theta\left(n^{\log_b^a}\right)$

2) If a= b^k

   a. If $p > -1$, then $T(n) = \Theta\left(n^{\log_b^a} \log^{p+1} n\right)$

   b. If $p = -1$, then $T(n) = \Theta\left(n^{\log_b^a} \log\log n\right)$

   c. If $p < -1$, then $T(n) = \Theta\left(n^{\log_b^a}\right)$

3) If $a < b^k$

   a. If $p \geq 0$, then $T(n) = \Theta(n^k \log^p n)$

   b. If $p < 0$, then $T(n) = O(n^k)$

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

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

$\Theta(n \log n)$

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

$a = 2,\ b = 2,\ k = 2,\ p = 0$

$2 < 4$

$\Theta(n^2 \log^0 n)$

$= \Theta(n^2)$

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

$3T\left(\frac{n}{2}\right) + n^2 \log^{-1} n$

$a = 3,\ b = 2,\ k = 2,\ p = -1$

$3 < 4$

$\Theta(n^2)$

1) If $a > b^k$, then $T(n) = \Theta\left(n^{\log_b^a}\right)$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \Theta\left(n^{\log_b^a} \log^{p+1} n\right)$
   b. If $p = -1$, then $T(n) = \Theta\left(n^{\log_b^a} \log\log n\right)$
   c. If $p < -1$, then $T(n) = \Theta\left(n^{\log_b^a}\right)$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \Theta(n^k \log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

Let $T(n) = T\left(\frac{n}{2}\right) + \frac{1}{2}n^2 + n$. What are the parameters?

Let $T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n} + 42$. What are the parameters?

Let $T(n) = 3T\left(\frac{n}{2}\right) + \frac{3}{4}n + 1$. What are the parameters?

Find the time complexity of the above recurrence functions

Thank you