Master Theorem
&
Solving Recurrence Relations

Prof. Shaik Naseera
CSE Department
JNTUA College of Engineering, Kalikiri

Objectives

- **Master Theorem**
- **Solving Recurrence Relations**
- **Discussion of Gate Questions**

Motivation: Asymptotic Behavior of Recursive Algorithms

- The time complexity of the algorithm is represented in the form of recurrence relation.
- When analyzing algorithms, recall that we only care about the asymptotic behavior
- Rather than solving exactly the recurrence relation associated with the cost of an algorithm, it is sufficient to give an asymptotic characterization
- The main tool for doing this is the master theorem

Master Theorem

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

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

  $$T(1) = c$$

  where $a \geq 1$, $b \geq 2$, $c>0$. If $f(n)$ is $\Theta(n^d)$ where $d \geq 0$ then

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{If } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

Master Theorem: Pitfalls

- You cannot use the Master Theorem if
  - T(n) is not monotone, e.g. T(n) = sin(x)
  - f(n) is not a polynomial, e.g., $T(n)=2T(n/2)+2^n$
  - b cannot be expressed as a constant, e.g.

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

- Note that the Master Theorem does not solve the recurrence equation
- Does the base case remain a concern?

Master Theorem: Example 1

- Let $T(n) = T(n/2) + ½\, n^2 + n$. What are the parameters?

  a = 1

  b = 2

  d = 2

  Therefore, which condition applies?

  $1 < 2^2$, case 1 applies

- We conclude that

$$T(n) \in \Theta(n^d) = \Theta\,(n^2)$$

Master Theorem: Example 2

- Let $T(n) = 2\,T(n/4) + \sqrt{n} + 42$. What are the parameters?

  a = 2

  b = 4

  d = 1/2

  Therefore, which condition applies?

  $2 = 4^{1/2}$, case 2 applies

- We conclude that

$$T(n) \in \Theta(n^d \log n) = \Theta(\log n \sqrt{n})$$

Master Theorem: Example 3

- Let T(n)= 3 T(n/2) + 3/4n + 1. What are the parameters?

  a = 3

  b = 2

  d = 1

  Therefore, which condition applies?

  $3 > 2^1$, case 3 applies

- We conclude that

$$T(n) \in \Theta(n^{\log_b a}) = \Theta(n^{\log_2 3})$$

- Note that $\log_2 3 \approx 1.584$..., can we say that $T(n) \in \Theta(n^{1.584})$

  No, because $\log_2 3 \approx 1.5849$... and $n^{1.584} \notin \Theta(n^{1.5849})$

- $T(n) = 2T(\sqrt{n}) + \log n$.
  - Let $n = 2^m \Rightarrow m = \log n$
  - Then $T(2^m) = 2T(2^{m/2}) + m$.
  - Now let $S(m) = T(2^m)$.
  - Then $S(m) = 2S(m/2) + m$.
  - This is case-2 of master theorem and has the solution
  - $S(m) = O(m \log m)$.
  - So $T(n) = T(2^m)$
  - $\Rightarrow S(m) = O(m \log m) = O(\log n \log \log n)$.

Example

What is the value of following recurrence.

$T(n) = 5T(n/5) + \sqrt{n},$

$T(1) = 1,$

$T(0) = 0$

(A) Theta (n)

(B) Theta (n^2)

(C) Theta (sqrt(n))

(D) Theta (nLogn)

a=5, b=5, d=1/2

$a>b^d$ =>Theta ($n^{\log_5 5}$)=Theta(n)

Answer: (A)

Master Theorem

- **4th Condition**

‘Fourth’ Condition

- Recall that we cannot use the Master Theorem if f(n), the non-recursive cost, is not a polynomial
- There is a limited 4^{th} condition of the Master Theorem that allows us to consider poly logarithmic functions
- **Corollary**: If $f(n) \in \Theta(n^{\log_b a} \log^k n)$ for some k≥0 then

$$T(n) \in \Theta(n^{\log_b a} \log^{k+1} n)$$

‘Fourth’ Condition: Example

- Say we have the following recurrence relation

$$T(n) = 2\,T(n/2) + n \log n$$

- Clearly, a=2, b=2, but f(n) is not a polynomial. However, we have $f(n) \in \Theta(n \log n)$, k=1
- Therefore by the 4th condition of the Master Theorem we can say that

$$T(n) \in \Theta(n^{\log_b a} \log^{k+1} n) = \Theta(n^{\log_2 2} \log^2 n) = \Theta(n \log^2 n)$$

More Examples of Master's Theorem

- T(n) = 3T(n/5) + n $\Theta(n)$
- T(n) = 2T(n/2) + n $\Theta(n\log n)$
- T(n) = 2T(n/2) + 1 $\Theta(n)$
- T(n) = T(n/2) + n $\Theta(n)$
- T(n) = T(n/2) + 1 $\Theta(\log n)$

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{If } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

where $a \geq 1$, $b \geq 2$, $c>0$. If $f(n)$ is $\Theta(n^d)$ where $d \geq 0$

GATE QUESTIONS

Q. NO. 1

GATE CSE 1999

If $T_1 = O(1)$, give the correct matching for the following pairs:

List - I

(M) $T_n = T_{n-1} + n$

(N) $T_n = T_{n/2} + n$

(O) $T_n = T_{n/2} + n\log n$

(P) $T_n = T_{n-1} + \log n$

List - II

(U) $T_n = O(n)$

(V) $T_n = O(n\log n)$

(W) $T_n = O(n^2)$

(X) $T_n = O(\log^2 n)$

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{If } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

where $a \geq 1$, $b \geq 2$, $c>0$. If $f(n)$ is $\Theta(n^d)$ where $d \geq 0$

- A M – W N – V O – U P – X
- B M – W N – U O – X P – V
- C M – V N – W O – X P – U
- D M – W N – U O – V P – X

Answer : None of the above

(M)

$$T(n) = T(n-1) + n$$
$$= T(n-2) + (n-1) + n$$
$$= T(n-3) + (n-2) + (n-1) + n$$
$$\vdots$$
$$= T(n-k) + (n-(k-1)) + \cdots + n$$

Let $n-k=1$

$$= T(1) + 2 + 3 + \cdots + n$$
$$= 1 + 2 + \cdots n$$
$$= \frac{n(n+1)}{2} = O(n^2)$$

(N).

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

$a=1 \quad b=2 \quad d=1$

$a<b \quad \Theta(n)$

$$= O(n)$$

- (O). T(n)=T(n/2)+nlogn   Let=> $n=2^m$
- $=T(2^{m-1})+2^m.m$
- Let $S(m)=T(2^m)$
- $S(m) = S(m-1)+m\ 2^m$
- $=S(m-2)+(m-1)2^{m-1}+m2^m$
- $=S(m-k)+(m-(k-1))2^{m-(k-1)}+m2^m$
- $=S(1)+2.\ 2^0+3.\ 2^1+\ldots.+m.\ 2^m$
- $\leq C.\ m.\ 2^m= O(m.2^m)=O(\log n.\ n)=O(n.\log n)$

(P).

T(n) =T(n-1)+log n

=[T(n-2)+log(n-1)]+log n

=[T(n-3)+log(n-2)]+log(n-1)+log n

= .

.

.

=T(n-k)+log(n-(k-1))+.....+log n

=T(1)+log 2+log 3+....+log n

=T(1)+log(1)+log 2+....+log n //log 1 is 0

=log(1.2.3....n)

=log(n!) (note: n! upper bound is n^n)

=O(nlogn)

Q. NO.2

GATE CSE 2009

The running time of an algorithm is represented by the following recurrence relation:

$$T(n) = \begin{cases} n & n \leq 3 \\ T(\frac{n}{3}) + cn & \text{otherwise} \end{cases}$$

Which one of the following represents the time complexity of the algorithm?

- (A) $\Theta(n)$
- (B) $\Theta(n \log n)$
- (C) $\Theta(n^2)$
- (D) $\Theta(n^2 \log n)$

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{If } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

where $a \geq 1$, $b \geq 2$, $c>0$. If f(n) is $\Theta(n^d)$ where $d \geq 0$

Case I of master theorem

Q. NO.3

GATE CSE 2006

Consider the following recurrence:

$T(n) = 2T(\lceil\sqrt{n}\rceil)+1 \qquad T(1) = 1$

Which one of the following is true?

(A) $T(n) = \theta(\text{loglogn})$

(B) $T(n) = \theta(\text{logn})$

(C) $T(n) = \theta(\sqrt{n})$

(D) $T(n) = \theta(n)$

Substitute $n=2^m \Rightarrow T(2^m)=2T(2^{m/2})+1$

$\Rightarrow T(2^m)=2T(2^m /2)+1$

let $S(m)=T(2^m)$

$S(m)=2S(m/2)+1$

Q. NO. 4

GATE CSE 2005

Suppose T(n) = 2T (n/2) + n, T(0) = T(1) = 1

Which one of the following is FALSE?

- A. $T(n) = O(n^2)$
- B. $T(n) = \theta(n \log n)$
- C. $T(n) = \Omega(n^2)$
- D. $T(n) = O(n \log n)$

$T(n)=n\log n$

$n\log n <= O(n^2)$, O represent upper bound, True

θ represent both lower & upper bound

$C1 n\log n <= n\log n <= c2\ n\log n$

Ω represent lower bound, $n^2 <= n\log n$, false

Q. NO.5

GATE CSE 1997

Let T(n) be the function defined by $T(1) = 1,\ T(n) = 2T(\lfloor \frac{n}{2} \rfloor) + \sqrt{n}$

Which of the following statements is true?

- A $T(n) = O\sqrt{n}$
- B $T(n) = O(n)$
- C $T(n) = O(\log n)$
- D $T(n) = O(\log n)$

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{If } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

where $a \geq 1$, $b \geq 2$, $c>0$. If $f(n)$ is $\Theta(n^d)$ where $d \geq 0$

Case 3 is true, a=2, b=2, d=1/2

Q. NO.6

GATE CSE 1996

The recurrence relation

T(1) = 2

T(n) = 3T(n/4) + n

has the solution, T(n) equals to

- (A) O(n)
- (B) O(log n)
- (C) O(n^3/4)
- (D) None of the above

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{If } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

where $a \geq 1$, $b \geq 2$, $c>0$. If $f(n)$ is $\Theta(n^d)$ where $d \geq 0$

Case 1 is true, a=3, b=4, d=1

Q. NO.7

GATE CSE 2015 Set 2

An unordered list contains n distinct elements. The number of comparisons to find an element in this list that is neither maximum nor minimum is

A $\Theta(n \log n)$

B $\Theta(n)$

C $\Theta(\log n)$

D $\Theta(1)$

because all elements are distinct, select any three numbers and output 2nd largest from them.

Q. NO.8

GATE CSE 2008

The most efficient algorithm for finding the number of connected components in an undirected graph on n vertices and m edges has time complexity

A $\Theta(n)$

B $\Theta(m)$

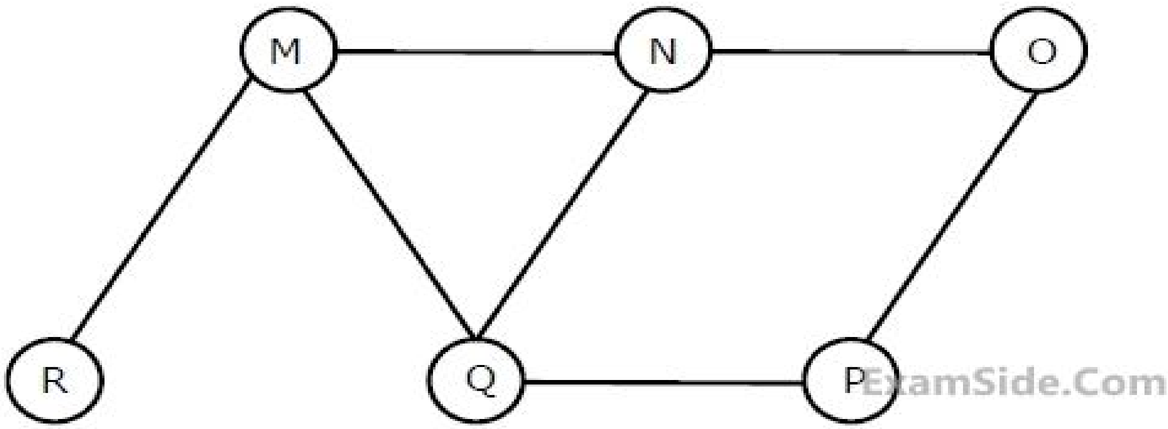
C $\Theta(n+m)$

D $\Theta(mn)$

Q. NO.9

GATE CSE 2008

The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is



A MNOPQR

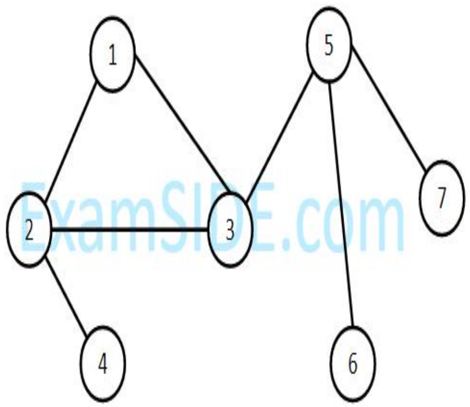
B NQMPOR

C QMNPRO

D QMNPOR

GATE CSE 1999

The number of articulation points of the following graph is



A 0

B 1

C 2

D 3

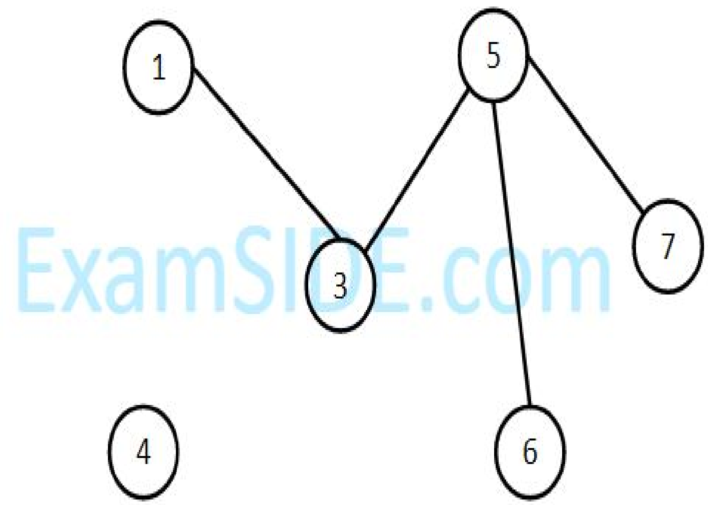
Explanation

Articulation Point is a point in a graph if we remove that point from the graph then the graph gets disconnected.

2, 3 and 5 are articulation point in this graph.

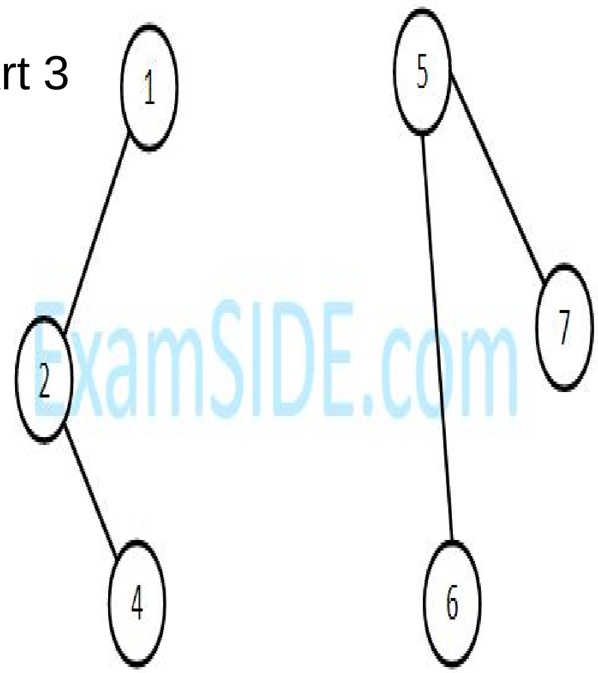
When we remove the vertex 2 from the graph, vertex 4 gets disconnected. See the below diagram.

Art 2



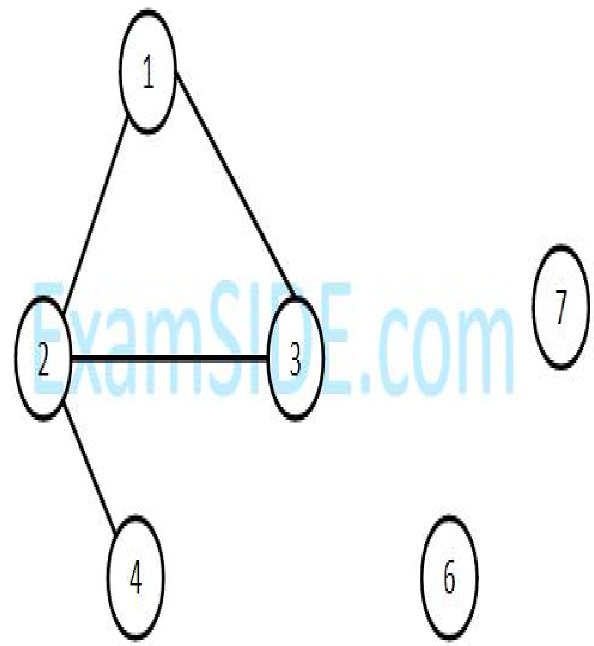
When we remove the vertex 3 from the graph, two differnt subgraph created. See the below diagram.

Art 3



When we remove the vertex 5 from the graph, vertex 6 and 7 gets disconneted from the graph. See the below diagram.

Art 5



Q. NO. 10

GATE | GATE-CS-2002 | Question 3

The solution to the recurrence equation $T(2^k) = 3\ T(2^{k-1}) + 1$, $T(1) = 1$, is:

(A) 2^k

(B) $(3^{k+1} - 1)/2$

(C) $3^{\log_2 k}$

(D) $2^{\log_3 k}$

$T(2^k) = 3\ T(2^{k-1}) + 1$

$= 3^2\ T(2^{k-2}) + 1 + 3$

$= 3^3\ T(2^{k-3}) + 1 + 3 + 9$

. . . (k steps of recursion (recursion depth))

$= 3^k\ T(2^{k-k}) + (1 + 3 + 9 + 27 + \ldots + 3^{k-1})$

$= 3^k + ((3^k - 1) / 2)$

$= ((2 * 3^k) + 3^k - 1)/2$

$= ((3 * 3^k) - 1) / 2$

$= (3^{k+1} - 1) / 2$

Hence, B is the correct choice.

Practice Problems

1. $T(n) = 3T(n/2) + n^2 \implies T(n) = \Theta(n^2)$ (Case 1)
2. $T(n) = 4T(n/2) + n^2 \implies T(n) = \Theta(n^2 \log n)$ (Case 2)
3. $T(n) = T(n/2) + 2^n \implies \Theta(2^n)$ Master Theorem not applicable, Possible with substitution method
4. $T(n) = 2^n T(n/2) + n^n \implies$ Does not apply (a is not constant)
5. $T(n) = 16T(n/4) + n \implies T(n) = \Theta(n^2)$ (Case 3)
6. $T(n) = 2T(n/2) + n \log n \implies T(n) = n \log^2 n$ (Case 2)
7. $T(n) = 2T(n/2) + n/\log n \implies$ Does not apply (non-polynomial difference between $f(n)$ and $n^{\log_b a}$)
8. $T(n) = 2T(n/4) + n^{0.51} \implies T(n) = \Theta(n^{0.51})$ (Case 1)
9. $T(n) = 0.5T(n/2) + 1/n \implies$ Does not apply ($a < 1$)
10. $T(n) = 16T(n/4) + n! \implies T(n) = \Theta(n!)$ (Case 1)
11. $T(n) = \sqrt{2}T(n/2) + \log n \implies T(n) = \Theta(\sqrt{n})$ (Case 3)
12. $T(n) = 3T(n/2) + n \implies T(n) = \Theta(n^{\lg 3})$ (Case 3)

Practice Problems

12. $T(n) = 3T(n/2) + n \Longrightarrow T(n) = \Theta(n^{\lg 3})$ (Case 3)

13. $T(n) = 3T(n/3) + \sqrt{n} \Longrightarrow T(n) = \Theta(n)$ (Case 3)

14. $T(n) = 4T(n/2) + cn \Longrightarrow T(n) = \Theta(n^2)$ (Case 3)

15. $T(n) = 3T(n/4) + n \log n \Longrightarrow T(n) = \Theta(n \log n)$ (Case 1)

16. $T(n) = 3T(n/3) + n/2 \Longrightarrow T(n) = \Theta(n \log n)$ (Case 2)

17. $T(n) = 6T(n/3) + n^2 \log n \Longrightarrow T(n) = \Theta(n^2 \log n)$ (Case 1)

18. $T(n) = 4T(n/2) + n/\log n \Longrightarrow T(n) = \Theta(n^2)$ (Case 3)

19. $T(n) = 64T(n/8) - n^2 \log n \Longrightarrow$ Does not apply ($f(n)$ is not positive)

20. $T(n) = 7T(n/3) + n^2 \Longrightarrow T(n) = \Theta(n^2)$ (Case 1)

21. $T(n) = 4T(n/2) + \log n \Longrightarrow T(n) = \Theta(n^2)$ (Case 3)

THANK YOU