Practice Set 1

January 24, 2019

ASYMPTOTIC ANALYSIS

① For the functions n^k and c^n, what is the asymptotic relation between them? $k >= 1$, $c > 1$.

② For the functions $lg(n)$ and $\log_8(n)$, what is the asymptotic relationship between these?

③ Asymptotic relationship between $n^3 \log_2(n)$ and $3n \log_8(n)$?

④ Is $8^n = \Omega(4^n)$?

⑤ True or False: $\log_2(n^{\log_2 17})$ is $\Theta(\log_2(17^{\log_2 n}))$.

ANSWERS:

① n^k is $O(c^n)$ ② $lg(n)$ is $\Theta(\log_8(n))$

③ $n^3 \log_2(n)$ is $\Omega(3n \log_8(n))$ ④ True ⑤ True

SOLVE THE FOLLOWING USING TREE AND SUBSTITUTION METHOD

① $T(n) = T\left(\frac{n}{2}\right) + 1$; $T(1) = 3$

② $T(0) = 6$; $T(n) = T(n-1) + 2$

③ $T(1) = c$; $T(n) = T\left(\frac{n}{2}\right) + 2n$

④ $T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + cn^2$; $T(1) = 1$

⑤ $T(n) = 2T(n-1) + \Theta(1)$

⑥ $T(n) = \sqrt{n}\, T(\sqrt{n}) + n$ (Any Method)

⑦ $T(n) = \dfrac{1}{2 - T(n-1)}$; $T(0) = 0$ (Any Method)

ANSWERS

① $\Theta(\log n)$

② $\Theta(n)$

③ $\Theta(n \cancel{\log n})$

④ $O(n^2)$

⑤ $O(2^n)$

⑥ $\Theta(n \log \log (n))$

⑦ $T(n) = \dfrac{n}{n+1}$

Master Theorem:
Practice Problems and Solutions

Master Theorem

The Master Theorem applies to recurrences of the following form:

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

where $a \geq 1$ and $b > 1$ are constants and $f(n)$ is an asymptotically positive function.

There are 3 cases:

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a} \log^k n)$ with[1] $k \geq 0$, then $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ with $\epsilon > 0$, and $f(n)$ satisfies the regularity condition, then $T(n) = \Theta(f(n))$.
   Regularity condition: $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n.

Practice Problems

For each of the following recurrences, give an expression for the runtime $T(n)$ if the recurrence can be solved with the Master Theorem. Otherwise, indicate that the Master Theorem does not apply.

1. $T(n) = 3T(n/2) + n^2$

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

3. $T(n) = T(n/2) + 2^n$

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

5. $T(n) = 16T(n/4) + n$

6. $T(n) = 2T(n/2) + n \log n$

[1] most of the time, $k = 0$

7. $T(n) = 2T(n/2) + n/\log n$

8. $T(n) = 2T(n/4) + n^{0.51}$

9. $T(n) = 0.5T(n/2) + 1/n$

10. $T(n) = 16T(n/4) + n!$

11. $T(n) = \sqrt{2}T(n/2) + \log n$

12. $T(n) = 3T(n/2) + n$

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

14. $T(n) = 4T(n/2) + cn$

15. $T(n) = 3T(n/4) + n \log n$

16. $T(n) = 3T(n/3) + n/2$

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

18. $T(n) = 4T(n/2) + n/\log n$

19. $T(n) = 64T(n/8) - n^2 \log n$

20. $T(n) = 7T(n/3) + n^2$

21. $T(n) = 4T(n/2) + \log n$

22. $T(n) = T(n/2) + n(2 - \cos n)$

Solutions

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

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

3. $T(n) = T(n/2) + 2^n \implies \Theta(2^n)$ (Case 3)

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 1)

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 3)

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 3)

11. $T(n) = \sqrt{2}T(n/2) + \log n \implies T(n) = \Theta(\sqrt{n})$ (Case 1)

12. $T(n) = 3T(n/2) + n \implies T(n) = \Theta(n^{\lg 3})$ (Case 1)

13. $T(n) = 3T(n/3) + \sqrt{n} \implies T(n) = \Theta(n)$ (Case 1)

14. $T(n) = 4T(n/2) + cn \implies T(n) = \Theta(n^2)$ (Case 1)

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

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

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

18. $T(n) = 4T(n/2) + n/\log n \implies T(n) = \Theta(n^2)$ (Case 1)

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

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

21. $T(n) = 4T(n/2) + \log n \implies T(n) = \Theta(n^2)$ (Case 1)

22. $T(n) = T(n/2) + n(2 - \cos n) \implies$ Does not apply. We are in Case 3, but the regularity condition is violated. (Consider $n = 2\pi k$, where k is odd and arbitrarily large. For any such choice of n, you can show that $c \geq 3/2$, thereby violating the regularity condition.)

Infix to Post-fix conversion

1. $3 + 4 \times 5 \div 6$
2. $(300 + 23) \times (43 - 21) \div (84 + 7)$
3. $(4 + 8) \times (6 - 5) \div ((3 - 2) \times (2 + 2))$
4. $((AX + (B \times CY))/(D - E))$

Answers

1. $3\ 4\ 5 \times 6 \div +$
2. $300\ 23\ +\ 43\ 21 - \times 84\ 7 + \div$
3. $4\ 8 + 6\ 5 - \times 3\ 2 - 2\ 2 + \times \div$
4. $AX\ B\ CY \times + DE - \div$

Post-fix expression evaluation

1. $10\ 2\ 8\ \times + 3 -$
2. $100\ 200 + 2 \div 5 \times 7 +$
3. $6\ 5 \times 7\ 3 - 4\ 8 + \times +$
4. $5\ 3 + 6\ 2 \div \times 3\ 5 \times +$

Answers

1. 23
2. 757
3. 78
4. 39

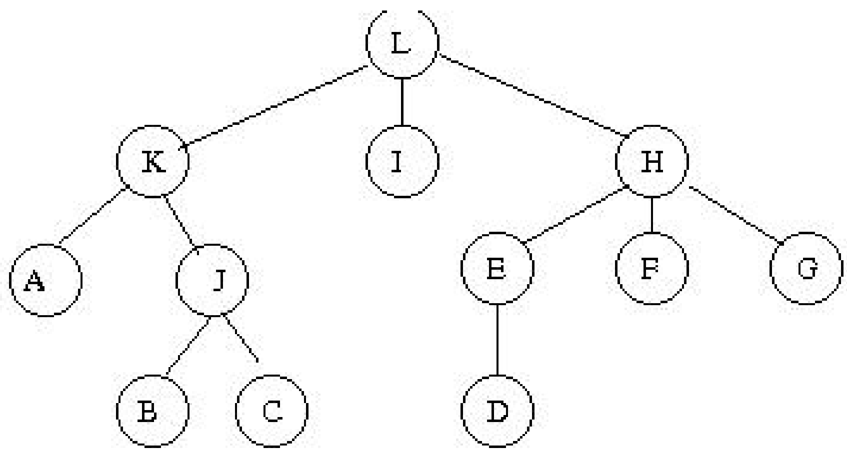
Binary Trees and BST

1. The height of a binary tree is the maximum number of edges in any root to leaf path. The maximum number of nodes in a binary tree of height h is?

2. Level of a node is distance from root to that node. For example, level of root is 1 and levels of left and right children of root is 2. The maximum number of nodes on level i of a binary tree is?

3. A complete n-ary tree is a tree in which each node has n children or no children. Let I be the number of internal nodes and L be the number of leaves in a complete n-ary tree. If L = 41, and I = 10, what is the value of n?

4. Write Preorder and Postorder traversal sequence for the following tree:



5. True or false: In a preorder traversal of a binary search tree, the first item printed out is always the smallest one. If true, explain why; if false, give an example where it is false.

6. T is a binary tree of height 3. What is the largest number of nodes that T can have? What is the smallest number?

7. T is a min heap of height 3. What is the largest number of nodes that T can have? What is the smallest number?

8. In a full binary tree if number of internal nodes is I, then number of leaves L are?

9. In a full binary tree if number of internal nodes is I, then number of nodes N are?

10. What are the worst case and average case complexities of a binary search tree for searching an element?

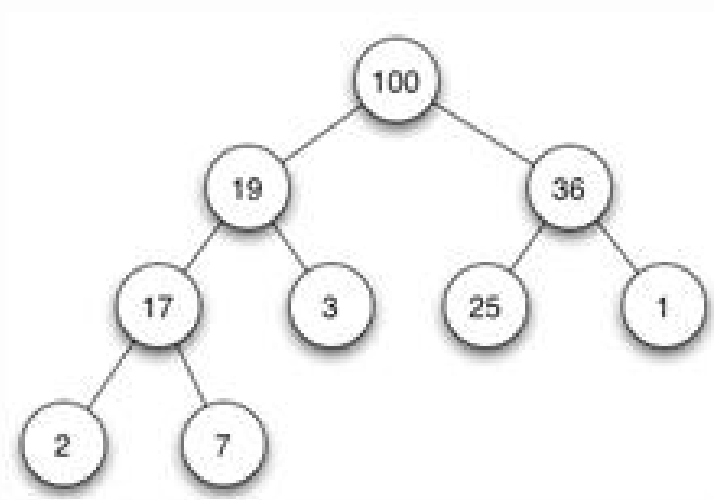
11. What is the space complexity of the in-order traversal in the recursive fashion? (d is the tree depth and n is the number of nodes)

12. In a max-heap, element with the greatest key is always in the which node?

13. The worst case complexity of deleting any arbitrary node value element from heap is?

14. An array consist of n elements. We want to create a heap using the elements. The time complexity of building a heap will be in order of?

15. If we implement heap as maximum heap , adding a new node of value 15 to the left most node of right subtree . What value will be at leaf nodes of the right subtree of the heap?



16. Which of the following statement about binary tree is CORRECT?

(a) Every binary tree is either complete or full (b) Every complete binary tree is also a full binary tree (c) Every full binary tree is also a complete binary tree (d) A binary tree cannot be both complete and full

17. Let b_n denote the number of different binary trees with n nodes. Find b_n?

18. We are given a set of n distinct elements and an unlabeled binary tree with n nodes. In how many ways can we populate the tree with the given set so that it becomes a binary search tree?

19. A binary search tree is used to locate the number 43. Which of the following probe sequences are possible and which are not? (A)61 52 14 17 40 43 (B)2 3 50 40 60 43 (C)10 65 31 48 37 43 (D)81 61 52 14 41 43 (E)17 77 27 66 18 43

Answers

1. $2^{(h+1)} - 1$

2. 2^{i-1}

3. 5

4. Preorder: L,K,A,J,B,C,I,H,E,D,F,G
   Postorder: A,B,C,J,K,I,D,E,F,G,H,L.

5. False

6. If every internal node of T has 2 children then there is one node at the root, 2 at depth 1, 4 at depth 2, 8 at depth 3, for a total of 15 nodes. If every internal node of T has 1 child, then there is a total of 4 nodes.

7. The first three levels (including the root) must be fully filled out, giving a total of 7 nodes. The 4th level has between 1 node and 8 nodes. So the total number of nodes in the heap is between 8 and 15.

8. $L = I + 1$

9. $N = 2I + 1$

10. $O(n), O(logn)$

11. $O(d)$
12. root node
13. $O(logn)$
14. $O(nlogn)$
15. 15 and 1
16. C
17. $\frac{^{2n}C_n}{n+1}$
18. 1
19. ACD