Data Structures Test Bank 2015

Data Structures Binary Trees

Question 1

Which of the following is a true about Binary Trees

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 No binary tree is both complete and full.

E None of the above

Question 1 Explanation:

A full binary tree (sometimes proper binary tree or 2-tree or strictly binary tree) is a tree in which every node other than the leaves has two children. A complete binary tree is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible. A) is incorrect. For example, the following Binary tree is neither complete nor full

```
     12
   /
  20
 /
30
```

B) is incorrect. The following binary tree is complete but not full

```
     12
   /   \
  20    30
 /
30
```

C) is incorrect. Following Binary tree is full, but not complete

```
     12
   /   \
  20    30
```

```
      / \
     20   40
```

D) is incorrect. Following Binary tree is both complete and full

```
     12
   /   \
  20    30
 /  \
10   40
```

Please refer http://en.wikipedia.org/wiki/Binary_tree#Types_of_binary_trees

Question 2

If arity of operators is fixed, then which of the following notations can be used to parse expressions without parentheses? a) Infix Notation (Inorder traversal of a expression tree) b) Postfix Notation (Postorder traversal of a expression tree) c) Prefix Notation (Preorder traversal of a expression tree)

A b and c

B Only b

C a, b and c

D None of them

Question 3

What are the main applications of tree data structure? 1) Manipulate hierarchical data 2) Make information easy to search (see tree traversal). 3) Manipulate sorted lists of data 4) Router algorithms 5) Form of a multi-stage decision-making, like Chess Game. 6) As a workflow for compositing digital images for visual effects

A 1, 2, 3, 4 and 6

B 1, 2, 3, 4 and 5

C 1, 3, 4, 5 and 6

D 1, 2, 3, 4, 5 and 6

Data Structures Test Bank 2015

Question 3 Explanation:

See http://en.wikipedia.org/wiki/Tree_(data_structure)#Common_uses

Question 4

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

In the following answers, the operator '^' indicates power.

A 2^(i-1)

B 2^i

C 2^(i+1)

D 2^[(i+1)/2]

Question 4 Explanation:

Answer: 2^(i-1) where i ≧1. Proof: by Induction, • Introduction base: i=1 (root) The number of node is: 2^(i-1) = 2^0 = 1. • Induction hypothesis Assume that for i ≧1, the maximum number of nodes on level i-1 is 2^(i-2). • Induction step Since each node in a binary tree has a maximum degree of 2. Therefore, the maximum number of nodes on level i is 2*2^(i-2) which is 2^(i-1)

Question 5

In a complete k-ary tree, every internal node has exactly k children or no child. The number of leaves in such a tree with n internal nodes is:

A nk

B (n – 1) k+ 1

C n(k – 1) + 1

D n(k – 1)

Question 5 Explanation:

For an k-ary tree where each node has k children or no children, following relation holds L = (k-1)*n + 1 Where L is the number of leaf nodes and n is the number of internal nodes. Let us see following for example

```
             o
        /    |    \
      o      o      o
   / | \          / | \
  o  o  o       o  o  o
                 / | \
                o  o  o


k = 3
Number of internal nodes n = 4
Number of leaf nodes = (k-1)*n  + 1
                     = (3-1)*4 + 1
                     = 9
```

Question 6

The maximum number of binary trees that can be formed with three unlabeled nodes is:

A 1

B 5

C 4

D 3

Question 6 Explanation: Following are all possible unlabeled binary trees

```
          O
       /     \
     O         O
        (i)


         O
       /
     O
   /
 O
     (ii)


      O
    /
  O
     \
       O
    (iii)


O
   \
     O
       \
         O
  (iv)


     O
       \
         O
       /
     O
  (v)
```

Note that nodes are unlabeled. If the nodes are labeled, we get more number of trees.

Question 7

The number of leaf nodes in a rooted tree of n nodes, with each node having 0 or 3 children is:

A n/2

B (n-1)/3

C (n-1)/2

D (2n+1)/3

Question 7 Explanation:

Let L be the number of leaf nodes and I be the number of internal nodes, then following relation holds for above given tree (For details, please see question 3 of http://geeksforgeeks.org/?p=4545)

```
L = (3-1)I + 1 = 2I + 1
```

Total number of nodes(n) is sum of leaf nodes and internal nodes

```
n = L + I
```

After solving above two, we get L = (2n+1)/3

Question 8

A weight-balanced tree is a binary tree in which for each node. The number of nodes in the left sub tree is at least half and at most twice the number of nodes in the right sub tree. The maximum possible height (number of nodes on the path from the root to the farthest leaf) of such a tree on n nodes is best described by which of the following? a) b) c) d)

A A

B B

C C

D D

Question 8 Explanation:

Let the maximum possible height of a tree with n nodes is represented by H(n). The maximum possible value of H(n) can be approximately written using following recursion

```
  H(n) = H(2n/3) + 1
```

The solution of above recurrence is [Tex]\log_{3/2} n[/Tex]. We can simply get it by drawing a recursion tree. **4. Consider the following algorithm for searching for a given number x in an unsorted - array A[1..n] having n distinct values:**

```
1)      Choose an i uniformly at random from 1..n;
2)      If A[i] = x then Stop else Goto 1;
```

Assuming that x is present in A, what is the expected number of comparisons made by the algorithm before it terminates? a) n b) n-l c) 2n d) n/2 Answer(a) If you remember the coin and dice questions, you can just guess the answer for the above. Below is proof for the answer. Let expected number of comparisons be E. Value of E is sum of following expression for all the possible cases.

```
number_of_comparisons_for_a_case * probability_for_the_case
```

Case 1

```
 If A[i] is found in the first attempt
 number of comparisons = 1
 probability of the case  = 1/n
```

Case 2

```
 If A[i] is found in the second attempt
 number of comparisons = 2
 probability of the case  = (n-1)/n*1/n
```

Case 3

```
 If A[i] is found in the third attempt
 number of comparisons = 2
 probability of the case  = (n-1)/n*(n-1)/n*1/n
```

There are actually infinite such cases. So, we have following infinite series for E.

```
E  = 1/n + [(n-1)/n]*[1/n]*2 + [(n-1)/n]*[(n-1)/n]*[1/n]*3 + ….  (1)
```

After multiplying equation (1) with (n-1)/n, we get

```
E (n-1)/n = [(n-1)/n]*[1/n] + [(n-1)/n]*[(n-1)/n]*[1/n]*2 +
                                 [(n-1)/n]*[(n-1)/n]*[(n-1)/n]*[1/n]*3 ……….(2)
```

Subtracting (2) from (1), we get

```
E/n = 1/n + (n-1)/n*1/n + (n-1)/n*(n-1)/n*1/n + …………
```

The expression on right side is a GP with infinite elements. Let us apply the sum formula (a/(1-r))

```
E/n = [1/n]/[1-(n-1)/n]  = 1

E = n
```

Question 9

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?

A 6

B 3

C 4

D 5

Question 9 Explanation:

For an n-ary tree where each node has n children or no children, following relation holds

```
    L = (n-1)*I + 1
```

Where L is the number of leaf nodes and I is the number of internal nodes. Let us find out the value of n for the given data.

```
L = 41 , I = 10

41 = 10*(n-1) + 1

(n-1) = 4
```

```
n = 5
```

Question 10

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:

A 2^h -1

B 2^(h-1) – 1

C 2^(h+1) -1

D 2*(h+1)

Question 10 Explanation:

Maximum number of nodes will be there for a complete tree. Number of nodes in a complete tree of height h = 1 + 2 + 2^2 + 2*3 + …. 2^h = 2^(h+1) – 1

Question 11

A scheme for storing binary trees in an array X is as follows. Indexing of X starts at 1 instead of 0. the root is stored at X[1]. For a node stored at X[i], the left child, if any, is stored in X[2i] and the right child, if any, in X[2i+1]. To be able to store any binary tree on n vertices the minimum size of X should be. (GATE CS 2006)

- **A** log2n
- **B** n
- **C** 2n + 1
- **D** 2^n — 1

Question 11 Explanation:

For a right skewed binary tree, number of nodes will be 2^n - 1. For example, in below binary tree, node 'A' will be stored at index 1, 'B' at index 3, 'C' at index 7 and 'D' at index 15.

```
A
 \
   \
    B
      \
        \
         C
           \
             \
              D
```

Question 12

Postorder traversal of a given binary search tree, T produces the following sequence of keys 10, 9, 23, 22, 27, 25, 15, 50, 95, 60, 40, 29 Which one of the following sequences of keys can be the result of an in-order traversal of the tree T? (GATE CS 2005)

A 9, 10, 15, 22, 23, 25, 27, 29, 40, 50, 60, 95

B 9, 10, 15, 22, 40, 50, 60, 95, 23, 25, 27, 29

C 29, 15, 9, 10, 25, 22, 23, 27, 40, 60, 50, 95

D 95, 50, 60, 40, 27, 23, 22, 25, 10, 9, 15, 29

Question 12 Explanation:

Inorder traversal of a BST always gives elements in increasing order. Among all four options, a) is the only increasing order sequence.

Question 13

Consider the following nested representation of binary trees: (X Y Z) indicates Y and Z are the left and right sub stress, respectively, of node X. Note that Y and Z may be NULL, or further nested. Which of the following represents a valid binary tree?

A (1 2 (4 5 6 7))

B (1 (2 3 4) 5 6) 7)

C (1 (2 3 4)(5 6 7))

D (1 (2 3 NULL) (4 5))

Question 13 Explanation:

C is fine.

```
(1 (2 3 4)(5 6 7))  represents following binary tree
      1
   /    \
  2      5
 / \    / \
3   4  6   7
```

A) (1 2 (4 5 6 7)) is not fine as there are 4 elements in one bracket. B) (1 (2 3 4) 5 6) 7) is not fine as there are 2 opening brackets and 3 closing. D) (1 (2 3 NULL) (4 5)) is not fine one bracket has only two entries (4 5)

Question 14

Consider a node X in a Binary Tree. Given that X has two children, let Y be Inorder successor of X. Which of the following is true about Y?

A Y has no right child

B Y has no left child

C Y has both children

D None of the above

Question 14 Explanation:

Since X has both children, Y must be leftmost node in right child of X.

Question 15

In a binary tree with n nodes, every node has an odd number of descendants. Every node is considered to be its own descendant. What is the number of nodes in the tree that have exactly one child?

A 0

B 1

C (n-1)/2

D n-1

Question 15 Explanation:

It is mentioned that each node has odd number of descendants including node itself, so all nodes must have even number of descendants 0, 2, 4 so on. Which means each node should have either 0 or 2 children. So there will be no node with 1 child. Hence 0 is answer. Following are few examples.

```
        a
     /     \
    b        c



       a
     /    \
    b       c
  /   \
 d      e
```

Such a binary tree is full binary tree (a binary tree where every node has 0 or 2 children).

Question 16

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:

A $2^h - 1$

B $2^{h-1} - 1$

C $2^{h+1} - 1$

D 2^{h+1}

Question 16 Explanation:

See Question 1 http://www.geeksforgeeks.org/data-structures-and-algorithms-set-10/

Question 17

The height of a tree is the length of the longest root-to-leaf path in it. The maximum and minimum number of nodes in a binary tree of height 5 are

A 63 and 6, respectively

B 64 and 5, respectively

C 32 and 6, respectively

D 31 and 5, respectively

Question 17 Explanation:

```
Number of nodes is maximum for a perfect binary tree.

A perfect binary tree of height h has 2^(h+1) - 1 nodes

Number of nodes is minimum for a skewed binary tree.
A perfect binary tree of height h has h+1 nodes.
```

Question 18

A binary tree T has 20 leaves. The number of nodes in T having two children is

A 18

B 19

C 17

D Any number between 10 and 20

Question 18 Explanation:

The number of nodes with two children is always one less than the number of leaves. See Handshaking Lemma and Interesting Tree Properties for proof.

Question 19

Consider a complete binary tree where the left and the right subtrees of the root are max-heaps. The lower bound for the number of operations to convert the tree to a heap is

A $\Omega(\log n)$

B $\Omega(n)$

C $\Omega(n \log n)$

D $\Omega(n^2)$

Question 19 Explanation:

It is simple call to max-heapify which recurses at most through height of heap.

مع تحيات مبرمجون بلا حدود 2015

تامر أبو الزينات 0786358466

محمود عماد 0785780440