GeeksforGeeks

A computer science portal for geeks

Placements Practice GATE CS IDE Q&A GeeksQuiz

Handshaking Lemma and Interesting Tree Properties

What is Handshaking Lemma?

Handshaking lemma is about undirected graph. In every finite undirected graph number of vertices with odd degree is always even. The handshaking lemma is a consequence of the degree sum formula (also sometimes called the handshaking lemma)

$$\sum_{v\in V} \deg(v) = 2|E|$$

How is Handshaking Lemma useful in Tree Data structure?

Following are some interesting facts that can be proved using Handshaking lemma.

1) In a k-ary tree where every node has either 0 or k children, following property is always true.

```
  L = (k - 1)*I + 1
Where L = Number of leaf nodes
      I = Number of internal nodes
```

Proof:

Proof can be divided in two cases.

Case 1 (Root is Leaf):There is only one node in tree. The above formula is true for single node as L = 1, I = 0.

Case 2 (Root is Internal Node): For trees with more than 1 nodes, root is always internal node. The above formula can be proved using Handshaking Lemma for this case. A tree is an undirected acyclic graph.

Total number of edges in Tree is number of nodes minus 1, i.e., |E| = L + I – 1.

All internal nodes except root in the given type of tree have degree k + 1. Root has degree k. All leaves have degree 1. Applying the Handshaking lemma to such trees, we get following relation.

```
  Sum of all degrees  = 2 * (Sum of Edges)
```

```
Sum of degrees of leaves +
Sum of degrees for Internal Node except root +
Root's degree = 2 * (No. of nodes - 1)

Putting values of above terms,
L + (I-1)*(k+1) + k = 2 * (L + I - 1)
L + k*I - k + I -1 + k = 2*L  + 2I - 2
L + K*I + I - 1 = 2*L + 2*I - 2
K*I + 1 - I = L
(K-1)*I + 1 = L
```

So the above property is proved using Handshaking Lemma, let us discuss one more interesting property.

2) In Binary tree, number of leaf nodes is always one more than nodes with two children.

```
   L = T + 1
Where L = Number of leaf nodes
      T = Number of internal nodes with two children
```

Proof:

Let number of nodes with 2 children be T. Proof can be divided in three cases.

Case 1: There is only one node, the relationship holds
as T = 0, L = 1.

Case 2: Root has two children, i.e., degree of root is 2.

```
Sum of degrees of nodes with two children except root +
Sum of degrees of nodes with one child +
Sum of degrees of leaves + Root's degree = 2 * (No. of Nodes - 1)

Putting values of above terms,
(T-1)*3 + S*2 + L + 2 = (S + T + L - 1)*2

Cancelling 2S from both sides.
  (T-1)*3 + L + 2 = (S + L - 1)*2
  T - 1 = L - 2
  T = L - 1
```

Case 3: Root has one child, i.e., degree of root is 1.

```
Sum of degrees of nodes with two children +
Sum of degrees of nodes with one child except root +
Sum of degrees of leaves + Root's degree = 2 * (No. of Nodes - 1)

Putting values of above terms,
T*3 + (S-1)*2 + L + 1 = (S + T + L - 1)*2

Cancelling 2S from both sides.
  3*T + L -1 = 2*T + 2*L - 2
  T - 1 = L - 2
  T = L - 1
```

Therefore, in all three cases, we get T = L-1.

We have discussed proof of two important properties of Trees using Handshaking Lemma. Many GATE questions have been asked on these properties, following are few links.

GATE-CS-2015 (Set 3) | Question 35
GATE-CS-2015 (Set 2) | Question 20
GATE-CS-2005 | Question 36
GATE-CS-2002 | Question 34
GATE-CS-2007 | Question 43

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Get Google Domain & Email

Google Hosted Email For Your Domain Start With A Free 30-Day Trial.

12 Comments Category: Trees

Related Posts:

- Check sum of Covered and Uncovered nodes of Binary Tree

- Lowest Common Ancestor in a Binary Tree | Set 2 (Using Parent Pointer)
- Construct a Binary Search Tree from given postorder
- BFS vs DFS for Binary Tree
- Maximum difference between node and its ancestor in Binary Tree
- Inorder Non-threaded Binary Tree Traversal without Recursion or Stack
- Check if leaf traversal of two Binary Trees is same?
- Closest leaf to a given node in Binary Tree

(Login to Rate and Mark)

2.2 Average Difficulty : **2.2/5.0**
Based on **5** vote(s)

Add to TODO List
Mark as DONE

Like Share 11 people like this.

Writing code in comment? Please use code.geeksforgeeks.org, generate link and share the link here.

12 Comments **GeeksforGeeks** 1 Login

Recommend 3 Share Sort by Newest

Join the discussion…

Hardik Agarwal • 3 months ago

The all above proofs can also be given in the following manner :

all nodes(external + internal) = 0*(no . of external) + 1* (no. of internal nodes with 1 child) + 2*(no. of internal nodes with 2 children) + ...+ 1(root)

• Reply • Share ›

Arun Sharma • 6 months ago

@geeksforgeeks I think the statement is incorrect "All internal nodes except root in the given type of tree have degree k + 1". The degree of anode is the number of its child. so the degree of the internal nodes should be at most k

• Reply • Share ›

shish_009 → Arun Sharma • 4 months ago

he is treating the tree as an undirected graph...handshaking lemma is applicable only for undirected graph

1 • Reply • Share ›

Shivam Anand • 8 months ago

I think the handshaking lemma unnescesarily complicates things. we can prove all of them using Mathematical induction. As an example, consider L=(k-1) *I + 1. The base case is when there is a single root. L=1 and I = 0. Plug it in the eqn, it is satisfied. Induction step is consider the case when we convert a leaf to an internal node. we add k leaves to this node.

L(new) = L(old) + k - 1. I(new) = I(old) +1. Plug it in, This too satisfies. Hence proved. Similarly others can be proved.

2 • Reply • Share ›

VAIBHAV GUPTA • 8 months ago

I think, explanation is given for full k-ary tree, bcoz a k-ary tree is a rooted tree in which each node has no more than k children and a full k-ary tree is a k-ary tree where within each level every node has either 0 or k children.

source: https://en.wikipedia.org/wiki/...

• Reply • Share ›

Avi Munjal • 8 months ago

How about a skewed tree. say root has one child and thats it.

In that case I=1 and L=1. Please explain where am I going wrong.

• Reply • Share ›

Ashish Singh → Avi Munjal • 7 months ago

In Binary tree, number of leaf nodes is always one more than nodes with two children. In your case, number of node with two child is 0. Number of leaf node is one. That's it.

• Reply • Share ›

ss • 9 months ago

(T-1)*3 + L + 2 = (S + L - 1)*2

T - 1 = L - 2

T = L - 1

here it would be t+l-1 on rhs

correct i t

@GeeksForGeeks

• Reply • Share ›

DS+Algo • 10 months ago

Another way to prove:

Let's suppose Initially we have only a root node as well as it would be leaf,

Suppose L=no. of leaf nodes and I=no. of Internal nodes

so initially I=0, L=1

Now, if we add k childs to that leaf node.

then this node becomes internal node so effectively (k-1) leaves are added and no. of internal nodes is incremented by1
hence, I=1, L=1 + (k-1)

Now, if we again add k childs to one of leaf nodes, then that would become internal so I increments by 1 and L increments by (k-1)
I=2, L=1 + (k-1) + (k-1)
.
.
.
I=n, L=1 + (k-1) I times
so L=1+I(k-1)

For binary tree, just substitute k=2
so L=1+I

• Reply • Share ›

Avatar

This comment was deleted.

DS+Algo Guest • 9 months ago

For binary tree L=T+1 where T is the no. Of nodes having two children. Nodes with 1 child do not affect no. Of leaves.

• Reply • Share ›

Avatar

This comment was deleted.

sahin Guest • 9 months ago

kya flaw hai is proof mein?

• Reply • Share ›

DS+Algo Guest • 9 months ago

Haha! Bhai maaf kar de.. abke kuch comment hi nhi krna bc

• Reply • Share ›

@geeksforgeeks, Some rights reserved Contact Us! About Us! Advertise with us!