**CS223 - Lesson 17 – worksheet on trees:**

Answer the following questions:

1. What is the root node? **A**
2. What are the children of node A? **B,C,D,E** Node E? **I,J,K** Node J? **None**
3. What are the siblings of node C? **B,D,E**  Node G? **F** Node K? **I,J**
4. What are the ancestors of node J? **E,A**  Node D? **A** Node H? **G,B,A**
5. What are the descendants of node D? **None** Node B? **G,H** Node A? **B->K**
6. Which nodes are terminal nodes? **F,H,C,D,I,J,K**
7. Which nodes are branch nodes? **A,B,G,E**
8. Given node E, list every relationship it has:  
   **Descendent of A, Sibling of B C D, Ancestor of I J K**
9. What is the degree of node B? **2** node D? **0** node A? **4**
10. What is the level of node A? **0** node G? **2** node H? **3**
11. What is the height of the tree? **4** Of the sub-tree whose root is E? **I,J,K**
12. Assuming the tree is a quad-tree:
    1. Is the tree full? **No**
    2. Is the tree complete? **No**
    3. Is the tree height-balanced? **No**
13. Define what it means that a tree has no cycles? **No nodes w/ multiple links that would lead to redundant iterations when navigating the tree**

Trees can hold a lot of data. Determine the amount of data in a binary tree at each level of the tree and for the entire tree to each level.

Nodes in Total nodes  
 Level: in tree

Level 0: **1** **1**   
  
Level 1: **2 3**  
  
  
  
Level 2: **4** **7**

Level 3: **8** **15**

Level n: **2^n 2^n!**

If you stored the name of every person on the earth, approximately 7 billion names, what would be the **height** of the binary tree?

**log2(7bill)=n n=33**

Perform the same analysis for a **ternary** tree. Determine a generalize formula for the number of nodes at each level of the tree and for the total tree based on the height of the tree.

Nodes in Lvl: Total Nodes:

Level 0: **1 1**

Level 1: **3 4**

Level 2: **6 10**

Level 3: **9 19**

Level n: **3^n 3^n + totalNodes(n-1)**

Perform the same analysis for a **quad** tree. Determine a generalize formula for the number of nodes at each level of the tree and for the total tree based on the height of the tree.

Nodes in Lvl: Total Nodes:

Level 0: **1 1**

Level 1: **4 5**

Level 2: **8 13**

Level 3: **16 29**

Level n: **4^n 4^n + totalNodes(n-1)**

Given a n-ary tree, how many possible nodes are on each level of the tree and in the total tree?

Nodes in Lvl: Total Nodes:

Level 0: **1 1**

Level 1: **n n+1**

Level 2: **n^2 n^2+n+1**

Level 3: **n^3 n^3+n^2+n+1**

Level k: **n^k n^k+ totalNodes(n-1)**

Describe a tree and draw a diagram of a tree that is equivalent to a list. We call this a *degenerate tree*.

**Only one child for each node, all on the same side**

Describe the advantages of a *complete* and a *height-balanced* tree.

**Easy to navigate, store, delete, etc. Minimum number of link traversals, minimal edges unused**

Tree *traversal* – process each node in a tree exactly once in a particular order.

* Given *n* nodes, there are *n*! (i.e., *n* factorial) possible orderings.
* For binary trees, we typically only care about 3 of the possible orders:

|  |  |  |
| --- | --- | --- |
| Pre-order traversal | In-order traversal | Post-order traversal |
| void preOrder(node) {  process(node)  preOrder(node->leftChild)  preorder(node->rightChild)  } | void inOrder(node) {  inOrder(node->leftChild)  process(node)  inOrder(node->rightChild)  } | void postOrder(node) {  postOrder(node->leftChild)  postOrder(node->rightChild)  process(node)  } |

Given this binary tree, what order will each of the traversals process the nodes?

Pre-order: **Top-Down recursion: A->B->D->H->I->E->J->K->C->F->L->M->G->N->O**

In-order: **H->D->I->B->J->E->K->A->L->F->M->C->N->G->O**

Post-order: **Bottom-Up recursion: H->I->D->J->K->E->B->L->M->F->N->O->G->C->A**

Figure out how you can "check your work" for the traversals.