VE 280 Lab 8

Out: 12:00, July 8, 2021; **Due**: 23:59, June 15, 2021.

Background

Note: You do not need to read this background to finish this lab, but you may find it helpful if you read it.

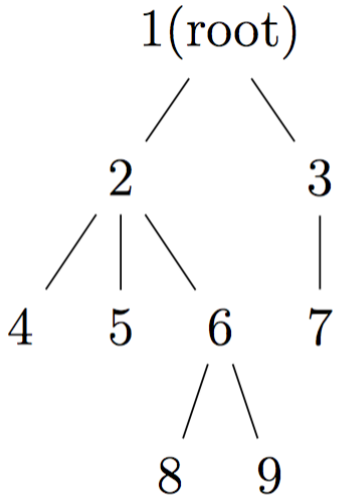
After experiencing *Blackjack* and *The Disappearance of Suzumiya Haruhi*, Kyon want to write down these two special experiences. However, the complicated game rule and the jumping timeline really puzzled Kyon. Below is from Kyon's monologue.

```
The game experience in November was really impressive. We chose to be players
with bankroll 100 and maximum hand of 20 to compete Stardust Crusaders. Koizumi
Itzuki chose to be a counting player, since he was really worried that Suzumiya
Haruhi might destroy the world if we lost the game. We played 6 rounds in total.
Suzumiya Haruhi beat Joseph Joestar in the first round, but she was kicked out by
Kujo Jotaro in the second round. Her dejection was a real delight to me. After
Nagato Yuki entered the game, she won in each round and kicked out all the
remaining members of Stardust Crusaders at last. I can only say, that's the way
she is.

Kyon > Do you still remember our playing process of Blackjack?
Yuki.N > There were in total 1402 lines of announcements in the game. The cards
were shuffled 23 times. Kujo Jotaro shouted 'Za Warudo' for 41 times. He
continuously shouted 'Za Warudo' for 9 times during the 9th hand with me. At
last, I got 139.
Kyon > Why did Kujo Jotaro shout 'Za Warudo'?
Yuki.N > Because he stopped the time and took the same number of cards again.
Kyon > Why didn't you point out that he cheated?
Yuki.N > Chengsong's fault. Cannot blame on me.

Who is Chengsong? I don't know. But Nagato is only an observer, so it is normal
for her to keep silent. But it is really inconceivable that someone other than
Haruhi can operate on time flow. It is really a miracle that we could beat
Stardust Crusade. If we do not have Nagato, we must have lost the game. I want to
write down this game experience, but the game rule is so complecated that I
myself did not really understand. How about drawing an N-ary tree to help me
figure out the rule?
```

Task 1: Implementation of N-ary Tree

Note: You should read this part carefully.

Your first task is to help `Kyon` implement the class of n-ary tree. A tree ADT organizes and manages data in a hierarchical tree structure. An n-ary tree is a generalization of a binary tree where any node in the tree has exactly one parent, except one node called root node, and any node may have zero up to n children, i.e. n is the **maximum number of child** for each node. For an example, see the following figure:



A tree ADT stores data in each node.

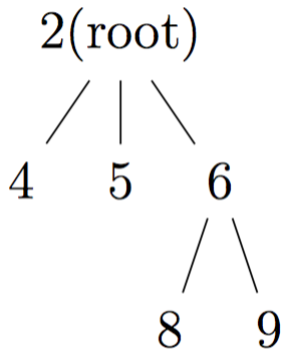
1.1 Terminology

1.1.1 Descendants

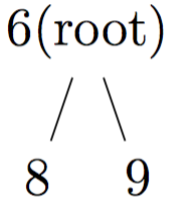
A `descendant` of a node X is a child of X or a descendant of a child of X.
For instance, the descendants of 2 in the previous figure are 4, 5, 6, 8, and 9.

1.1.2 Subtree

A `subtree` of the tree T is a tree consists of a node in T and all of this node's descendants.
For instance, if we name the tree above as T, then the tree rooted in 2 is a subtree of T.



Also, the the tree rooted in 6 is a subtree of T.



1.1.3 Leaf

A node with no child is called a `leaf` node. For example, node 4, 5, 8, 9 and 7 are leaf nodes.

1.1.4 Path

A `path` is a sequence of nodes such that a next node in the sequence is a child of the previous one.

For example 2->6->9 is a path and the length of this path is 2.

1.1.5 Height

The `height` of a node is the length of a longest path from the node to a leaf.

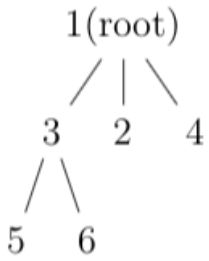
For example, height(1) = 3, height(9) = 0

1.2 Implementation

A node of a tree can be represented by the following class and a tree ADT can be represented by the node corresponding to its root.

```
class Node {
    // OVERVIEW: a node in the n-Ary tree, can also represent a n-ary tree rooted
at 'this'
protected:
    int value;      // the integer value of this
    int child_num;  // the number of child of this
    int n;          // n for this n-Ary tree
    Node *parent;   // parent node of this, for root node, parent = NULL
    Node **children;
    // children is an array of pointer to Node. Therefore, children is a pointer
of pointer
    int height;     // height of this node
    void addChild(Node *child);
    // REQUIRES: n of the child node is the same with n of this
    // EFFECTS: add the node child to the children array
    //          throw an exception tooManyChildren when child_num exceed n
    bool equal(Node* sub);
    // EFFECTS: return true if the tree rooted at 'this' and the tree rooted at
'sub'
    //          have the same shape and value.
public:
    Node(int _value, int _n = 2);
    // EFFECTS: create a root node with value and n

    ~Node();
    // EFFECTS: destroy the whole tree rooted at sub

    void addChild(int _value);
    // EFFECTS: create a child node with value and add it to the children array
    //          throw an exception tooManyChildren when child_num exceed n

    void traverse(vector<int>& traverseValue);
    // EFFECTS: insert the value of the nodes into traverseValue using a pre-
order traversal,
    //          A pre-order traversal insert the value of the node
    //          and then traverse its child nodes
    //          according to the sequence in children array.
    //          For example, the value of traverseValue of the tree above is
```

```
    //          1 2 4 5 6 8 9 3 7
    //          And the value of traverseValue of the tree below is
    //          1 3 5 6 2 4

    bool contain(Node *sub);
    // EFFECTS: return whether the tree rooted at sub is a subtree of this

    int getHeight();
    // EFFECTS: return height of this

    Node &operator[](int i);
    // EFFECTS: return a reference of (i+1) th child node of this,
    //          e.g. node1[0] returns a reference of the first child node of
node1
    //          if i is invalid, throw an invalidIndex
};
```

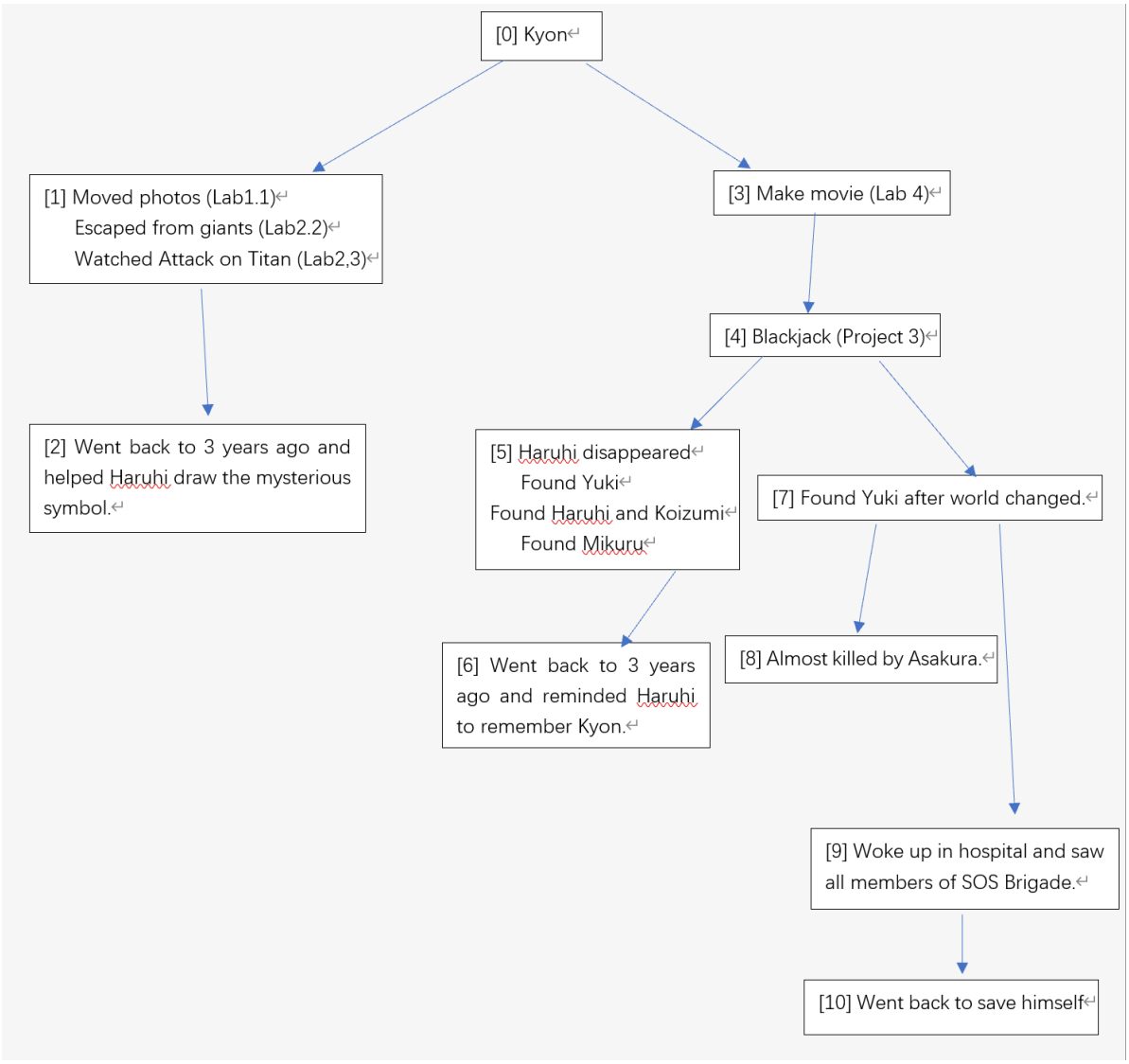


For this task, you are only allowed to implement the node into `node.cpp`, compress the file `node.cpp` and submit it to JOJ Lab 8 Task 1.

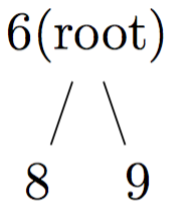
Task 2: Output the Tree of a Story in a Pre-order Way and Form a Full Story

Note: You do not need to read those string arrays and those story-related messages inside the graph if you don't want to. This is only to test whether your traverse function is correct. If it is, you will see a full story which should be logical. There will be only one test case for this task of which I will not show the detail to you.

Not long after *Blackjack*, in December, `Kyon` experienced a more outrageous incident: `Suzumiya Haruhi` disappeared from his life. The graph below briefly introduced the whole story of Kyon in the format of a n-ary tree.



Each number in the bracket is the index of the string vector `storyProcess`. Your task is to build a tree like this. The shape should be the same as above and the value in each node should be the same as the number in the bracket, for example, the value of the root node should be 0. Traverse it in a pre-order way, get the index sequence of the story and output the string vector `storyProcess` according to the index sequence you get. The index sequence for output is exactly the same as the pre-order traverse of the tree you build. For example, the pre-order traverse of the below tree is 6 8 9, then you output `storyProcess` according to the sequence `storyProcess[6]`, `storyProcess[8]`, `storyProcess[9]`. You do not need to add a blank space or a line break between each `storyProcess[i]`, but you need to add a line break at last.



The string vector `storyProcess` is given in `constant.h`.

For this task, you should put your main function in the file `storyTeller.cpp`, compress the file `storyTeller.cpp` and `node.cpp` into one zip/tar file and submit it to JOJ Lab 8 Task 2.

Testing

Remember to check memory leak by `valgrind`. Those who failed to avoid memory leak will only get half of the case grade. You will be provided `node.h` and `constant.h` files in the starter file.