Question 1 - 6 points

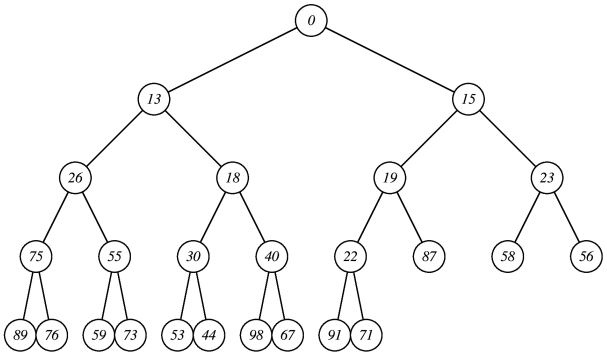
This was a multiple choice, and on canvas, the answers were shuffled. Here's an example:

If *f(n) = O(g(n))*, then:

- *A.* There exist positive constants *x* and *y* such that for all *n > x, y * f(n) > g(n)*.
- *B.* There exist positive constants *x* and *y* such that for all *n < x, y * g(n) > f(n)*.
- *C.* There exist positive constants *x* and *y* such that for all *n < x, y * f(n) > g(n)*.
- *D.* There exist positive constants *x* and *y* such that for all *n > x, y * g(n) > f(n)*.

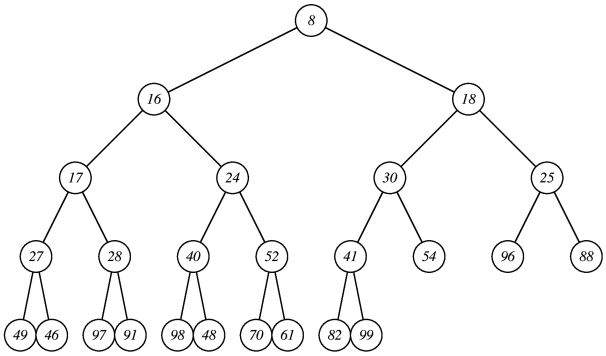
Question 2

These were from a bank of five similar, but different questions. Here's an example:



Suppose that Logan pushes the value 4 into the heap pictured above. When the resulting heap is stored in a vector named h, please tell me the following:

1. What is the value at index 1 of h (in other words, h.at(1))?
2. What is the value at index 2 of h (in other words, h.at(2))?
3. What is the value at index 5 of h (in other words, h.at(5))?
4. What is the value at index 6 of h (in other words, h.at(6))?
5. What is the value at index 11 of h (in other words, h.at(11))?
6. What is the value at index 12 of h (in other words, h.at(12))?
7. What is the value at index 24 of h (in other words, h.at(24))?
8. What is the value at index 25 of h (in other words, h.at(25))?



Suppose that Jacob pushes the value 12 into the heap pictured above.

When the resulting heap is stored in a vector named h, please tell me the following:

1. What is the value at index 1 of h (in other words, h.at(1)):
2. What is the value at index 2 of h (in other words, h.at(2)):
3. What is the value at index 5 of h (in other words, h.at(5)):
4. What is the value at index 6 of h (in other words, h.at(6)):
5. What is the value at index 11 of h (in other words, h.at(11)):
6. What is the value at index 12 of h (in other words, h.at(12)):
7. What is the value at index 24 of h (in other words, h.at(24)):
8. What is the value at index 25 of h (in other words, h.at(25)):

Questions 3-14

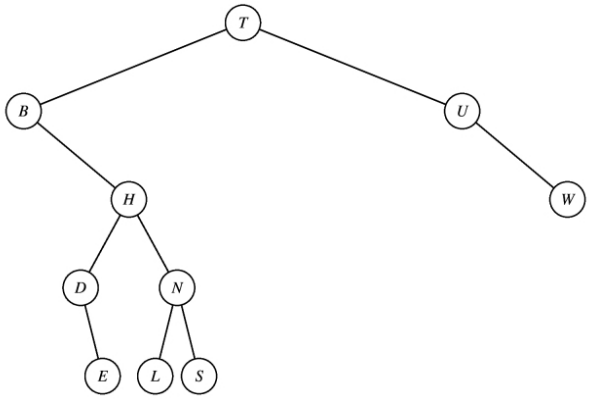
These came from a bank and were randomly ordered. Here they are:

- What is the Big-O running time of inserting the value 100 n times into an empty set of integers?
- What is the Big-O running time of inserting n integers in the front of a vector with n elements?
- What is the Big-O running time of finding a value in an AVL tree with n elements?
- Suppose that there are m unique elements in a multiset with n elements. What is the Big-O running time of printing all of those unique elements?
- What is the Big-O running time of creating a sorted vector from an AVL tree with n elements?
- What is the Big-O running time of given an iterator into a map with n elements, calling erase() on that iterator?
- What is the Big-O running time of creating a heap from a vector with n elements?
- What is the Big-O running time of removing the minimum element from a heap with n elements?
- What is the Big-O running time of dividing an integer n by 2 until it equals zero?
- What is the Big-O running time of inserting n values into an empty multimap?

- What is the Big-O running time of inserting n values repeatedly in front of the 3rd element in a list that starts with n elements (and n > 3)?
- What is the Big-O running time of inserting n values repeatedly in front of the 1st element in a deque with n elements?

Question 15

These came from a bank. Here's an example:

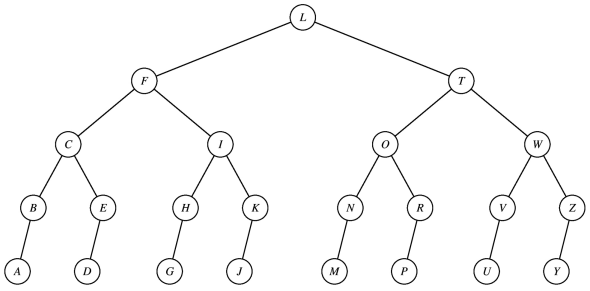


In the following blank, please enter a preorder printing of the nodes. You can just enter all of the letters, in order, without spaces:

And in the following blank, please enter a postorder printing of the nodes:

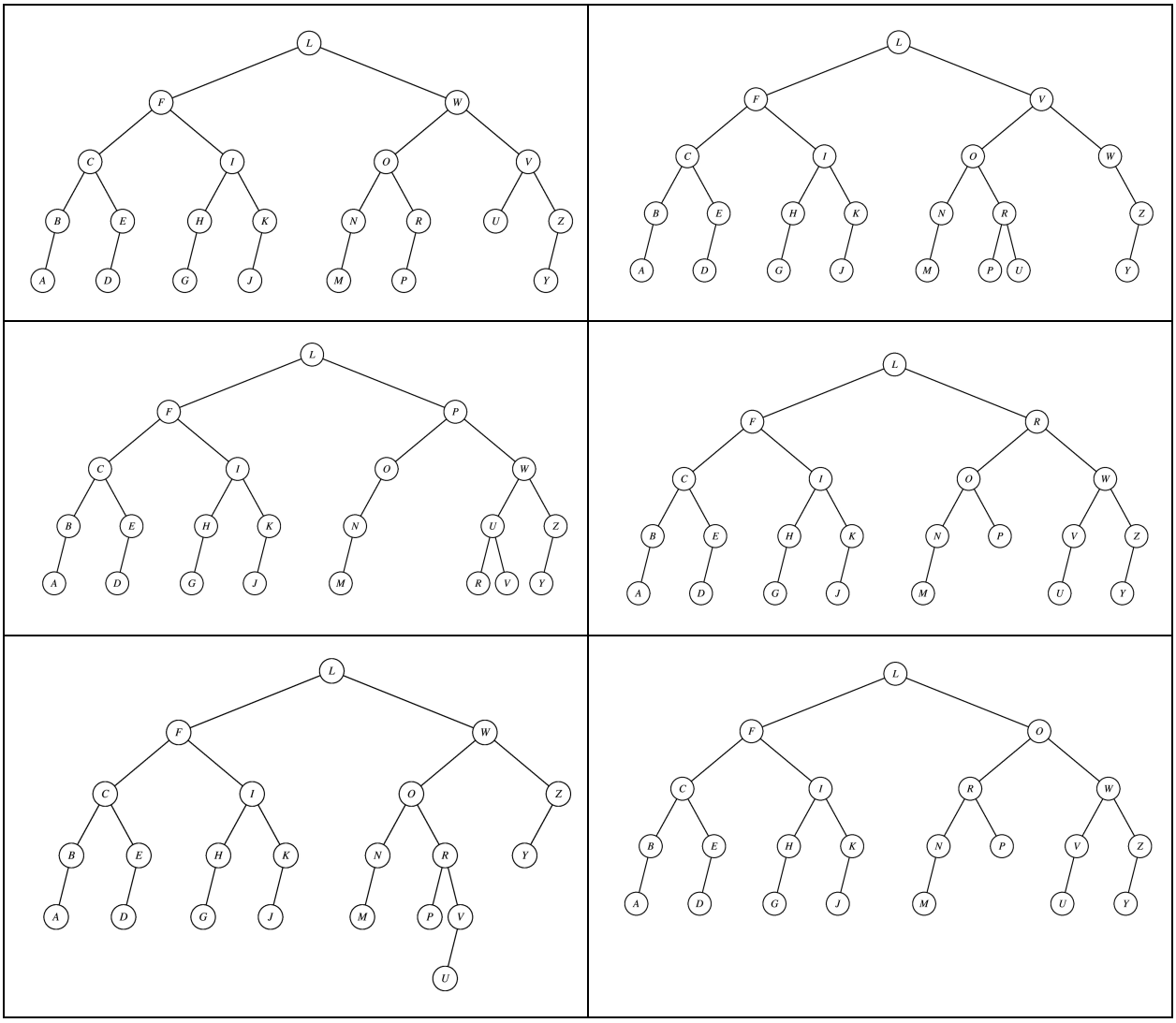
Question 16

These came from a bank. Here's an example:



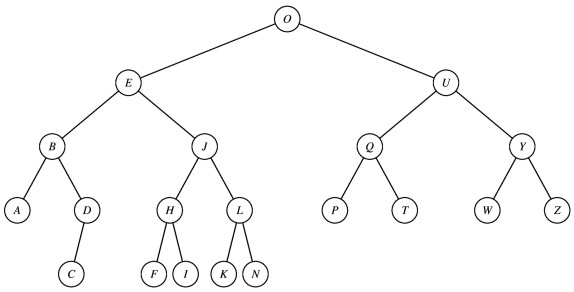
In the given binary search tree, which of the following trees results when you delete the node T?

This was a multiple-choice question.

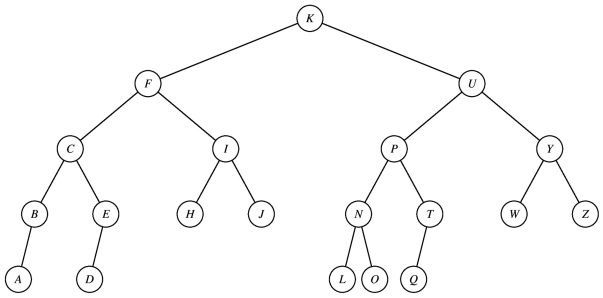


Questions 17 through 20

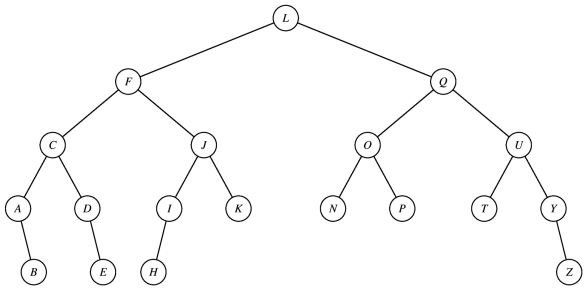
These were given in a random order:



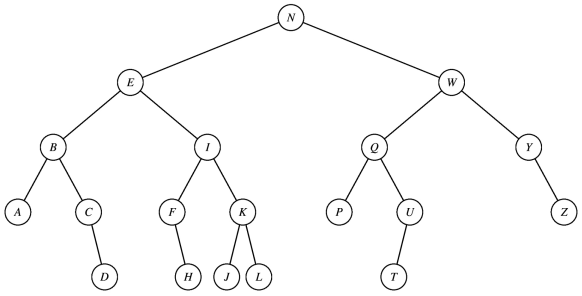
Suppose you insert M into the AVL tree above. The first rotation will be about the node _____. How many total rotations will there be? _____.



Suppose you insert M into the AVL tree above. The first rotation will be about the node _____. How many total rotations will there be? _____.



Suppose you delete T from the AVL tree above. The first rotation will be about the node _____. How many total rotations will there be? _____.



Suppose you delete Y from the AVL tree above. The first rotation will be about the node _____. How many total rotations will there be? _____.

Question 21

You have the following class definition in **zippy.hpp**:

```
class Zippy {
  public:
    Zippy();
    ~Zippy();
    Zippy(const Zippy &z);
    Zippy& operator= (const Zippy &z);

    Do_Stuff(const vector <int> &stuff);
    vector <int> Get_Stuff() const;
  protected:
    void *s;
};
```

In **zippy.cpp**, you have the following class definition:

```
class MyZippy {
  public:
    vector <int *> v;
};
```

Here's the constructor for Zippy:

```
Zippy::Zippy()
{
  MyZippy *z;

  z = new MyZippy;
  s = (void *) z;
};
```

I'm not going to show you any method implementations, but I'll tell you that Do_Stuff() makes multiple calls of the form "z->v.push_back(new int)". Please write the destructor. (Remember to set the entry box's "paragraph" to "preformatted").

Question 22

Recall the following class definitions from **dlist.hpp**:

```
class Dnode {
  friend class Dlist;
  public:
    std::string s;
    Dnode *Next();
    Dnode *Prev();
  protected:
    Dnode *flink;
    Dnode *blink;
};

class Dlist {
  public:
     /* A bunch of stuff that doesn't matter. */

    Dnode *Begin() const;       // Pointer to the first node on the list
    Dnode *End() const;         // Pointer to "one past" the last node on the list.

    void Insert_Before(const std::string &s, Dnode *n);

  protected:
    Dnode *sentinel;
    size_t size;
};
```

Implement these three methods. You cannot call any other methods of **Dlist**, and you cannot call the **Next()** or **Prev()** methods of the **Dnode** class.

Question 23

Below is the class definition of a node in a tree:

```
class TreeNode {
  public:
    vector <TreeNode *> children;
    double weight;
};
```

Let's define a "rank" of a subtree rooted by a node to be the maximum of the node's weight, and the average rank of the subtrees rooted by the node's children. Please write a procedure with the following prototype, that returns the rank of a node's subtree:

```
double rank(const TreeNode *n);
```

Question 24

Behold the following program:

```
#include <iostream>
#include <vector>
using namespace std;

void a()
{
  int i;
  vector <int> v;

  while (cin >> i) {
    v.push_back(i);
    if (i%2 == 0) a();
  }
  for (i = 0; i < v.size(); i++) cout << " " << v[i];
}

int main()
{
  a();
  cout << endl;
  return 0;
}
```

Suppose that this is compiled to a.out.In line one of your answer below, please put the output to:

```
echo 10 11 12 | ./a.out
```

In line two of your answer below, please put the output to:

```
echo 91 50 19 97 84 88 68 | ./a.out
```

Please answer in the proper format. Which means that line one contains the output of the first call, and line two contains the output of the second call. If you want to put other stuff in your answer, please put it after those first two lines.