Here is a simple class.

```
class Quiz2 {
public:
  Quiz2();
  Quiz2(const Quiz2& org);
  ~Quiz2();
  Quiz2& operator=(const Quiz2& rhs);
  // extracts contents of Quiz2 object from the input stream
  istream& extract(istream& os);
  // inserts contents of Quiz2 object (in sorted order) into the output stream
  ostream& insert(const ostream& os) const;
private:
  list<int> a;
};
```

Describe the errors with the following implementation of Quiz2::insert:

```
ostream& Quiz2::insert(const ostream& os) const
{
  a.sort();
  for(list<int>::const_iterator i=a.begin(); i<a.end(); ++i) {
    cout << *i << '\n';
  }
  return os;
}
```

Give the big-O time complexity for the following algorithm. Be sure to explain your reasoning.

```
void insertionSort (vector<double>& vec)
{
  for (int i = 1; i < vec.size(); i++) {
    // move element vec[i] into place
    int j = i - 1;
    while (j >= 0 && vec[j+1] < vec[j]) {
      // swap element
      double temp = vec[j];
      vec[j] = vec[j+1];
      vec[j+1] = temp;
      j--;                    // decrement j
    }
  }
}
```

Give the big-O time complexity for the following algorithm. Be sure to explain your reasoning.

```
void insertionSort (vector<double>& vec)
{
  for (int i = 1; i < vec.size(); i++) {
    // move element vec[i] into place
    int j = i - 1;
    while (j >= 0 && vec[j+1] < vec[j]) {
      // swap element
      double temp = vec[j];
      vec[j] = vec[j+1];
      vec[j+1] = temp;
      j--;                    // decrement j
    }
  }
}
```

Precisely and concisely explain why the subscript operator (operator[]) is not defined for the std::list class.

Create a STL stack, queue, and set and place the following two numbers into each of them: 15, 4.3.

You do not need to write a function or list the appropriate include files, just write the lines of code needed to do the above.

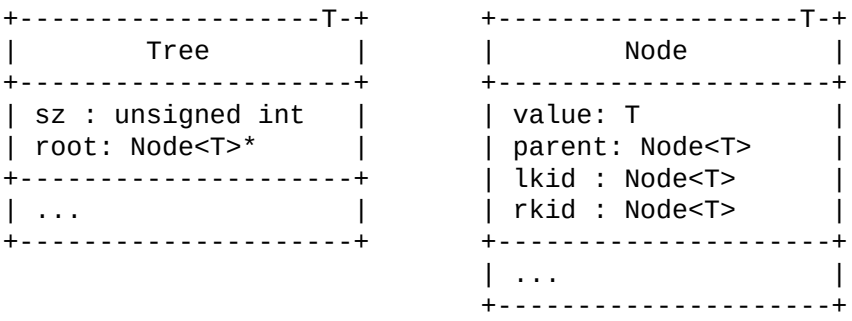
What does a hash function do?

For the Tree<T> class we began yesterday in lecture, write the

```
Tree<T>::insertRoot(const T& value);
```

member function. The function should assume that the tree is empty and add a root node with the value of value. You may not make use of any member functions in the Tree<T> class.

Recall we had:

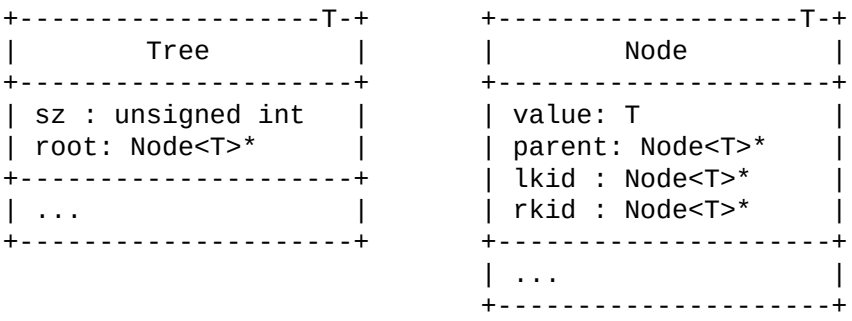


For the Tree<T> class we have been working on write the

```
Tree<T>::insertRoot(const T& value);
```

member function. The function should assume that the tree is empty and add a root node with the value of value. You may not make use of any member functions in the Tree<T> class.

Recall we had:



Write a function that will take two multimaps (`long unsigned int` index and `double` values), `x` and `y`, and return the inner product of the two sparse vectors. The inner product is defined as:

$$x \cdot y = \sum_{i=0}^{n-1} x_i y_i$$