Announcements

MP4 available, due 10/14, 11:59p. EC due 10/07, 11:59p.

Exam3: 10/16-10/19

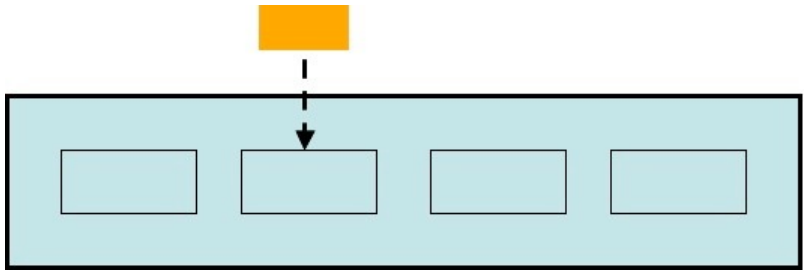
```
struct room {
   string name;
   int num;
   bool occupied;};

int main() {

   hotel<room> HI;
   // insert data into HI

   for(hotel<room>::iterator it = HI.begin(); it != HI.end(); it++)
      cout << (*it).num << "  " << (*it).occupant << endl;

   return 0;}
```



```
template <class T>
class hotel {
public:




private:



};
```

Where do these constructs live?

iterator class

begin()/end()

op++/op*

iter representation

std library documentation: http://www.sgi.com/tech/stl/

Generic programming: (more magic)

```
#include <list>
#include <iostream>
#include <string>
using namespace std;

struct animal {
   string name;
   string fo
   bool big;
   animal(st
};

int main() {

   animal g("
   list<anima

   zoo.push_back(g); zoo.push_back(p); zoo.push_back(b); //STL list insertAtEnd
   for(list<animal>::iterator it = zoo.begin(); it != zoo.end(); it++)
      cout << (*it).name << " " << (*it).food << endl;

return 0;
}
```

```
class printIfOcc {
public:
   void operator()(room a) {
      if (a.occupied) cout << a.num << a.name << endl;
   }
};
```

1. Declare an object of type `room`:
2. Declare an object of type `printIfOcc`:
3. Using your answers for 1 and 2, invoke a member function of the `printIfOcc` class:

Generic programming: (more magic)

```
template<class Iter, class Formatter>
void print(Iter first, Iter second, Formatter printer) {
   while (!(first==second)) {
      printer(*first);
      first++;
   }
}
```

Write a short description of this function:

This is a function called _____, whose inputs are two _____ and a _____. The function appears to _____

_____.

What is printer?

Generic programming: (more magic)

```
template<class Iter, class Formatter>
void print(Iter first, Iter second, Formatter printer) {
   while (!(first==second)) {
      printer(*first);
      first++;
   }
}
```

```
class printIfOcc {
public:
   void operator()(room a) {
      if (a.occupied) cout << a.num << a.name << endl;
   }
};
```

```
printIfOcc myFun;
print<hotel<room>::iterator,printIfOcc>(HI.begin(),HI.end(),myFun);
```

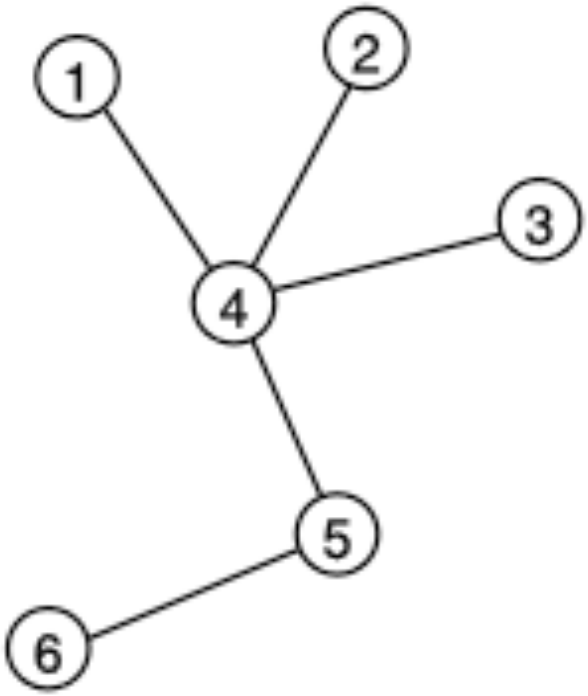
Trees:

"… most important nonlinear structure in computer science."

-- Donald Knuth, *Art of Computer Programming Vol 1*

A tree: _____

We'll study more specific trees:



Tree terminology:

• What's the longest English word you can make using the vertex labels in the tree (repeats allowed)?

• Find an edge that is not on the longest path in the tree. Give that edge a reasonable name.

For the rest of the exercises, assume the tree is rooted.

• One of the vertices is called the "root" of the tree. Guess which one it is.

• Make an English word containing the names of the vertices that have a parent but no sibling.

• How many parents does each vertex have?

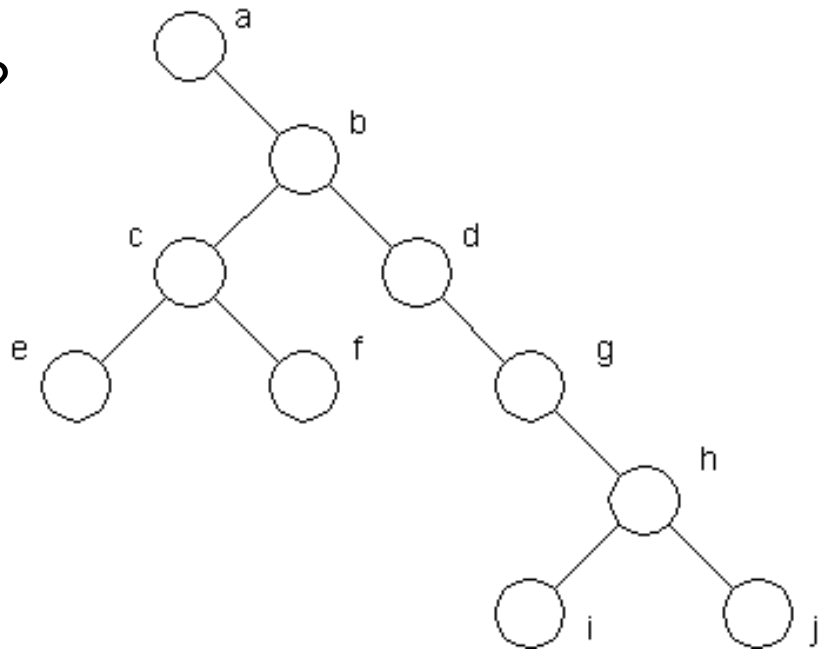
• Which vertex has the fewest children?

• Which vertex has the most ancestors?

• Which vertex has the most descendants?

• List all the vertices is b's left subtree.

• List all the leaves in the tree.



A rooted tree: