Announcements

MP4 available, due 3/8, 11:59p. EC due 3/1, 11:59p.

Code Challenge #1: Wed, 2/27, 9p, Siebel 0224.

TODAY: Queues - fin

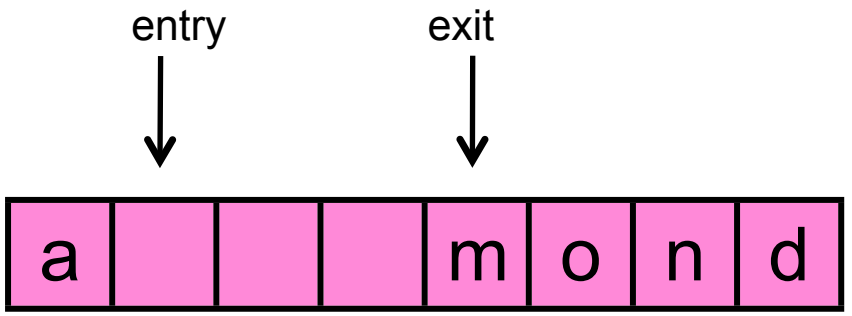
Intro to trees

Queue ADT:

- enqueue
- dequeue
- isEmpty

Queue array based implementation:

```
template<class SIT>
class Queue {
public:
    Queue();
    ~Queue(); // etc.
    bool empty() const;
    void enqueue(const SIT & e);
    SIT dequeue();
private:
    int capacity;
    int size;
    SIT * items;
    int entry;
    int exit;
    // some other stuff…
};
```



enqueue(y);

enqueue(i);

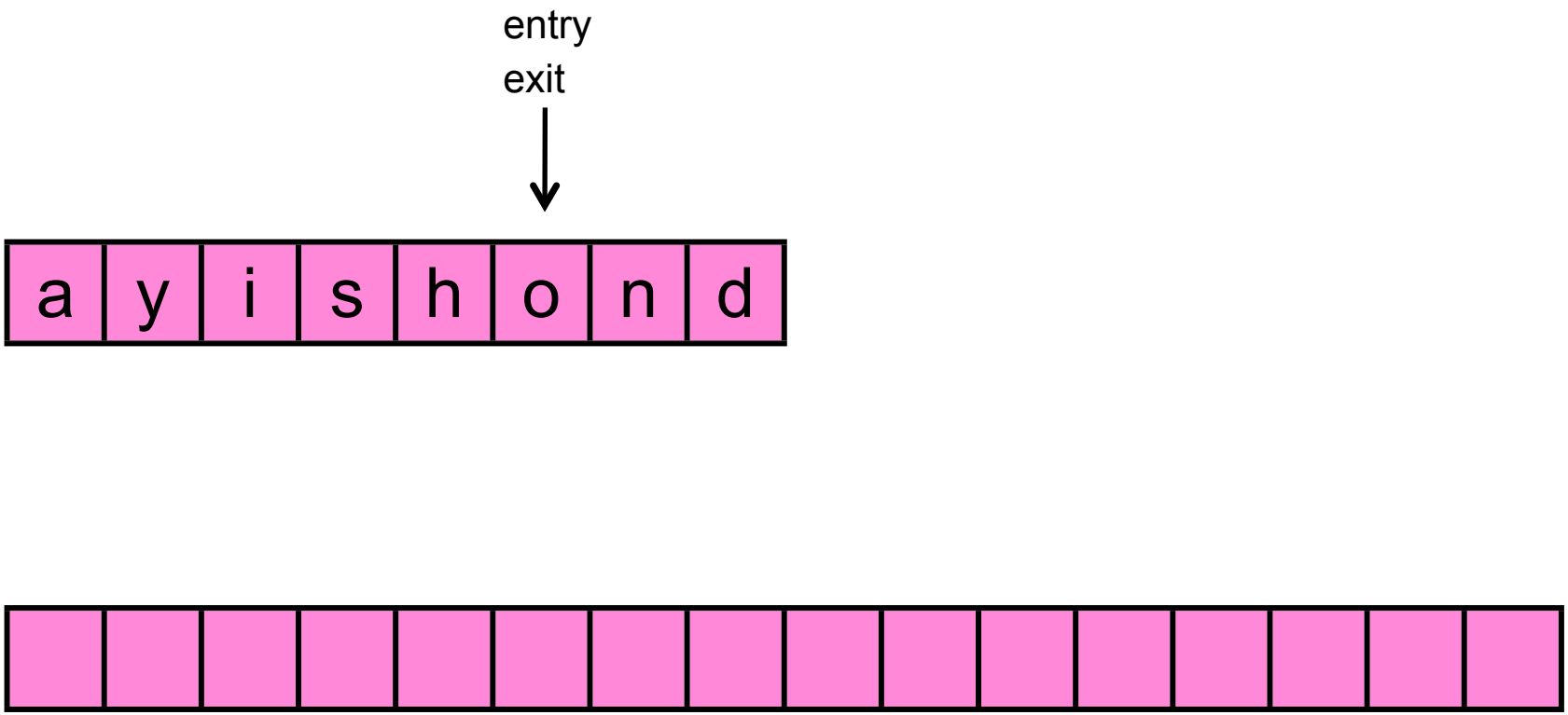
enqueue(s);

dequeue();

enqueue(h);

enqueue(a);

What if array fills?:



Another constrained access linear structure - Deque:

Deque ADT:

- pushFront
- pushRear
- popFront
- popRear

Reasonable Implementations:

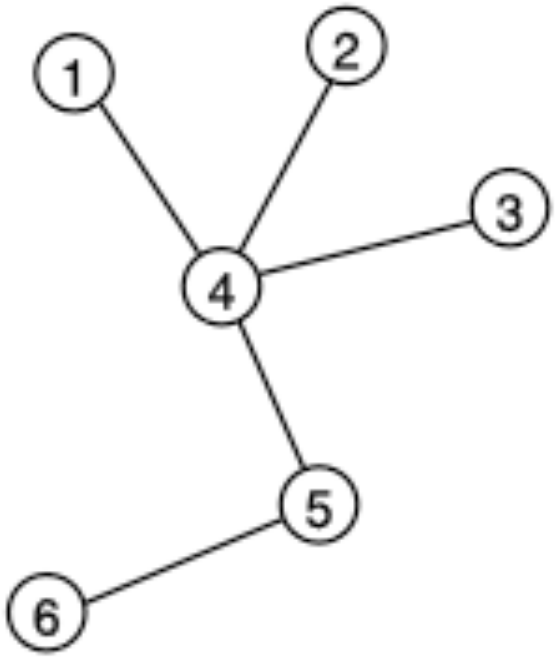
How does STL implement:

Trees:

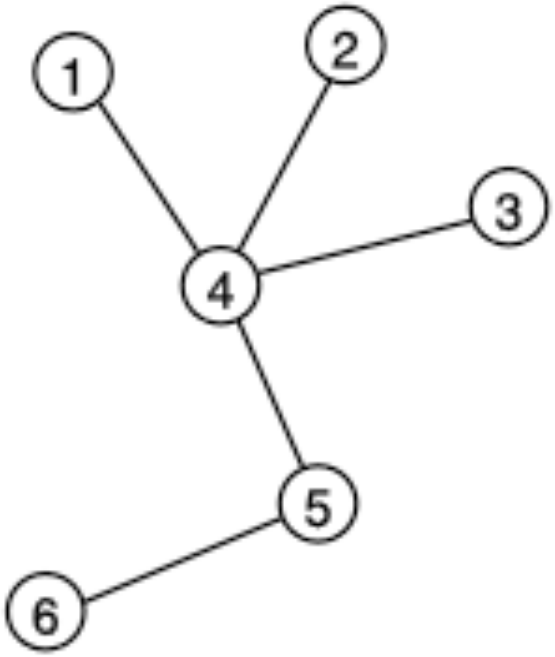
"… most important nonlinear structure in computer science."

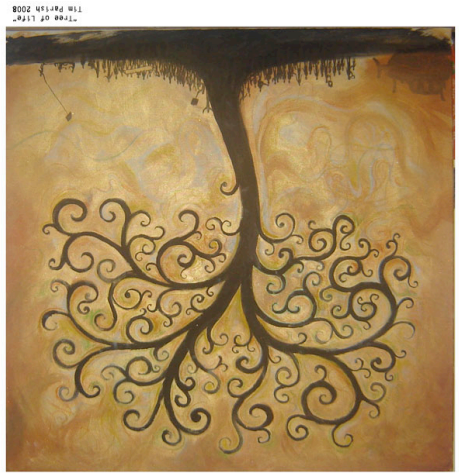
-- Donald Knuth, *Art of Computer Programming Vol 1*

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?

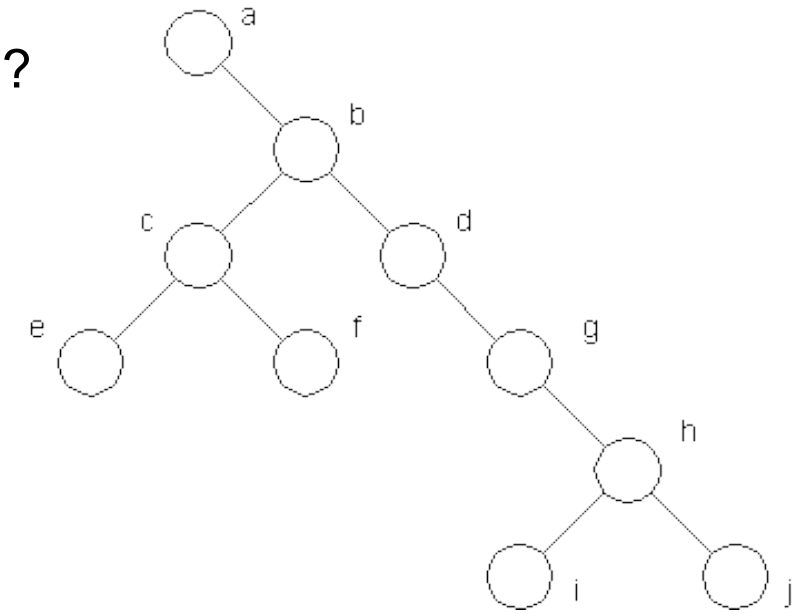
•Which vertex has the fewest children?

•Which vertex has the most ancestors?

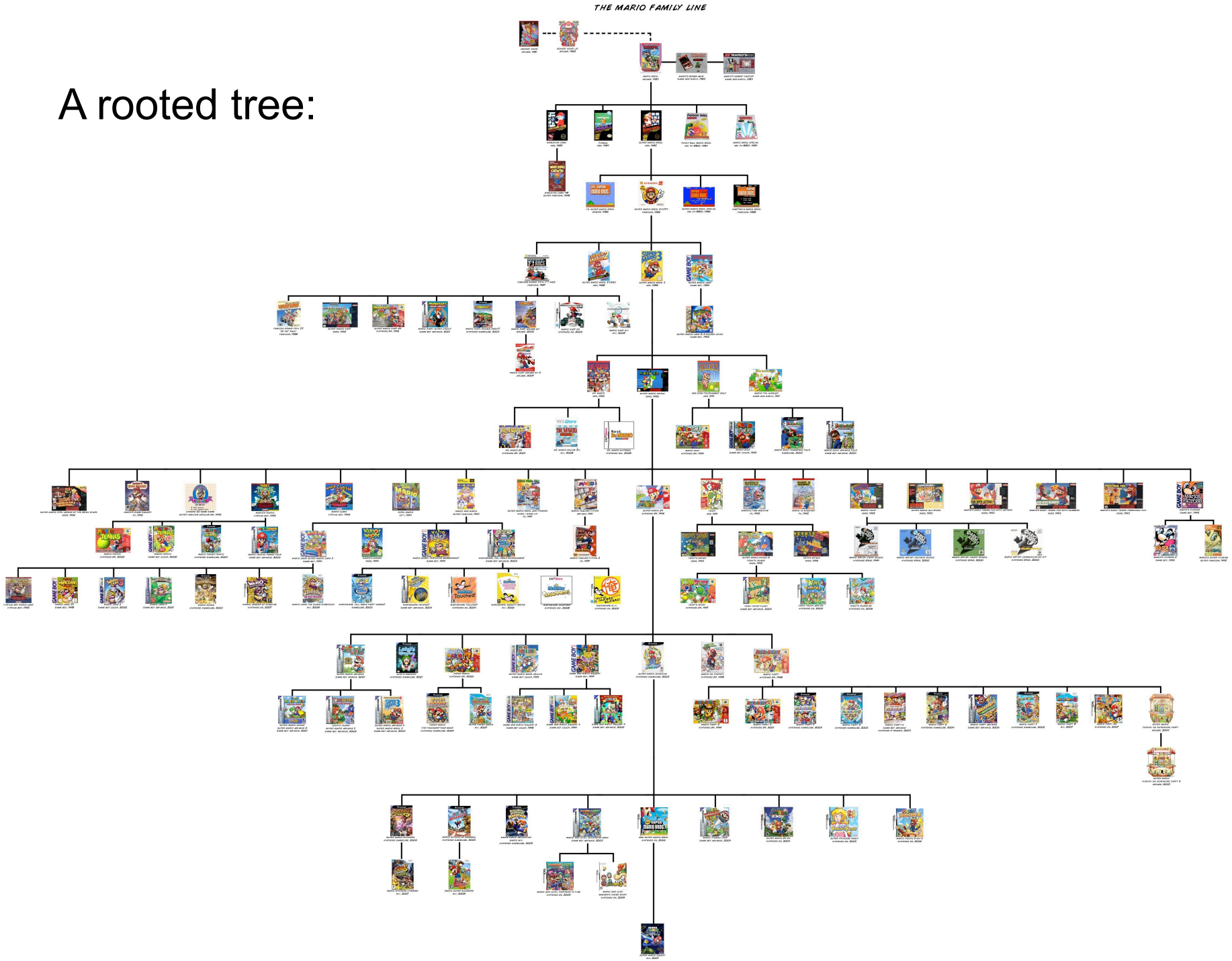
•Which vertex has the most descendants?

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

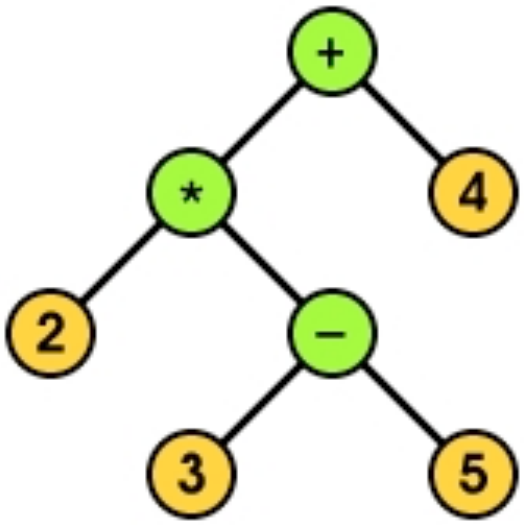
•List all the leaves in the tree.



A rooted tree:



Binary tree, recursive definition:



A *binary tree* T is either

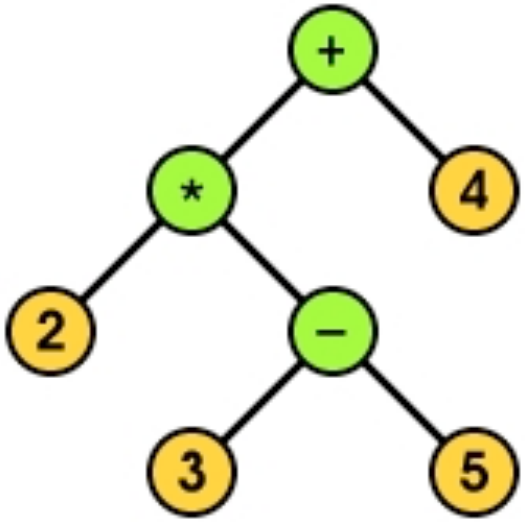
-

OR

-

An (important) example of a function on a binary tree:

height(t) -- length of longest path from root to a leaf



Given a tree T, write a recursive defn of the height of T, height(T):