ECE 2574
Introduction to Data Structures and Algorithms

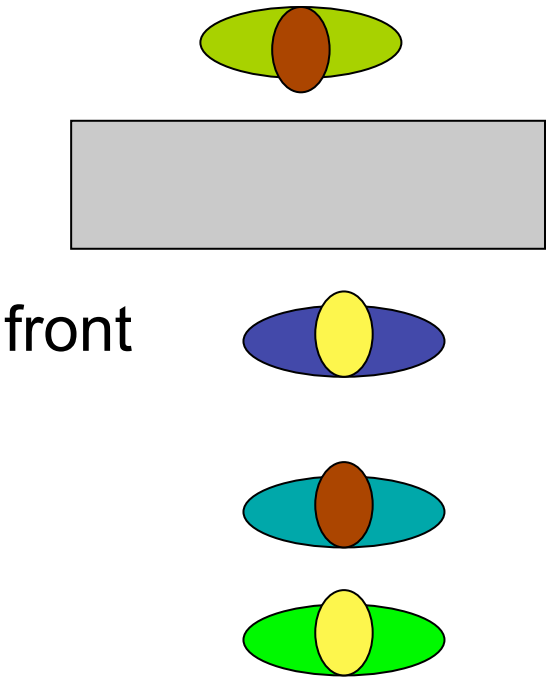
32: Heaps and Array-based Trees

Chris Wyatt
Electrical and Computer Engineering
Virginia Tech

Recall the definition of a queue

A variation is the introduction of a priority:

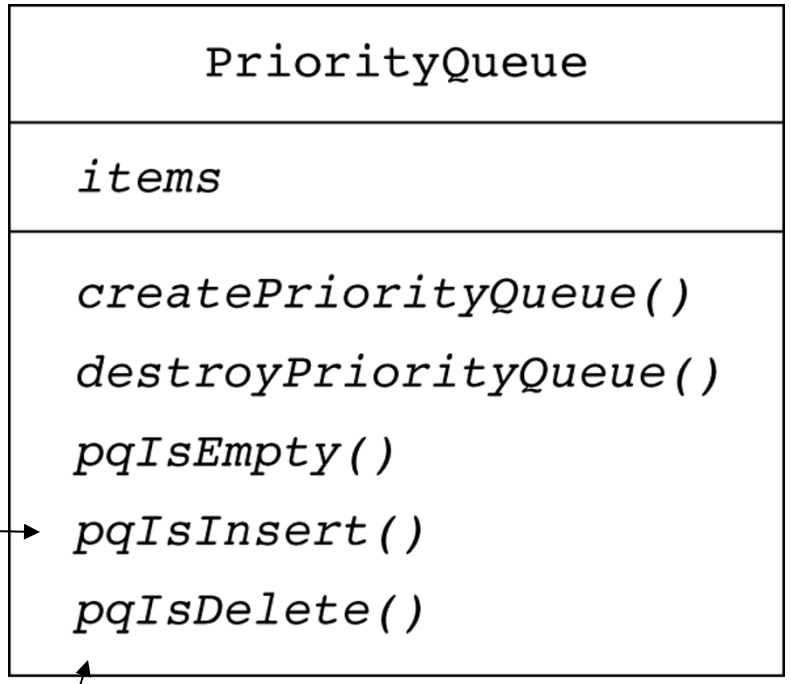
A **priority queue** or **heap**



Linked list or array implementations of a priority queue have O(n) performance for insert.

However a tree based implementation has O(log n) insertion.

Heap ADT



Insert is often called

push

Remove is often called

pop

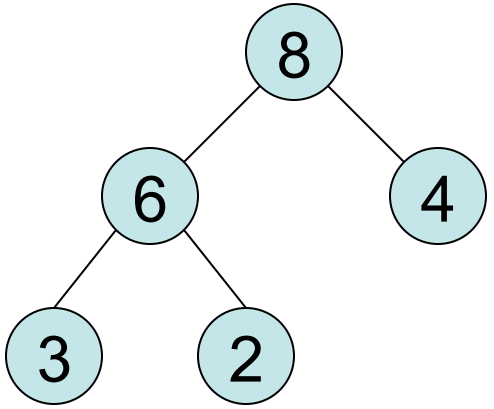
Just looking at the next record to be removed is sometimes called top

Heap implementation of priority queue

A (max) heap is a binary tree with the following properties.

- The tree is empty

or



-The root is the largest key

and

each subtree is also a heap

Note the tree is always complete.

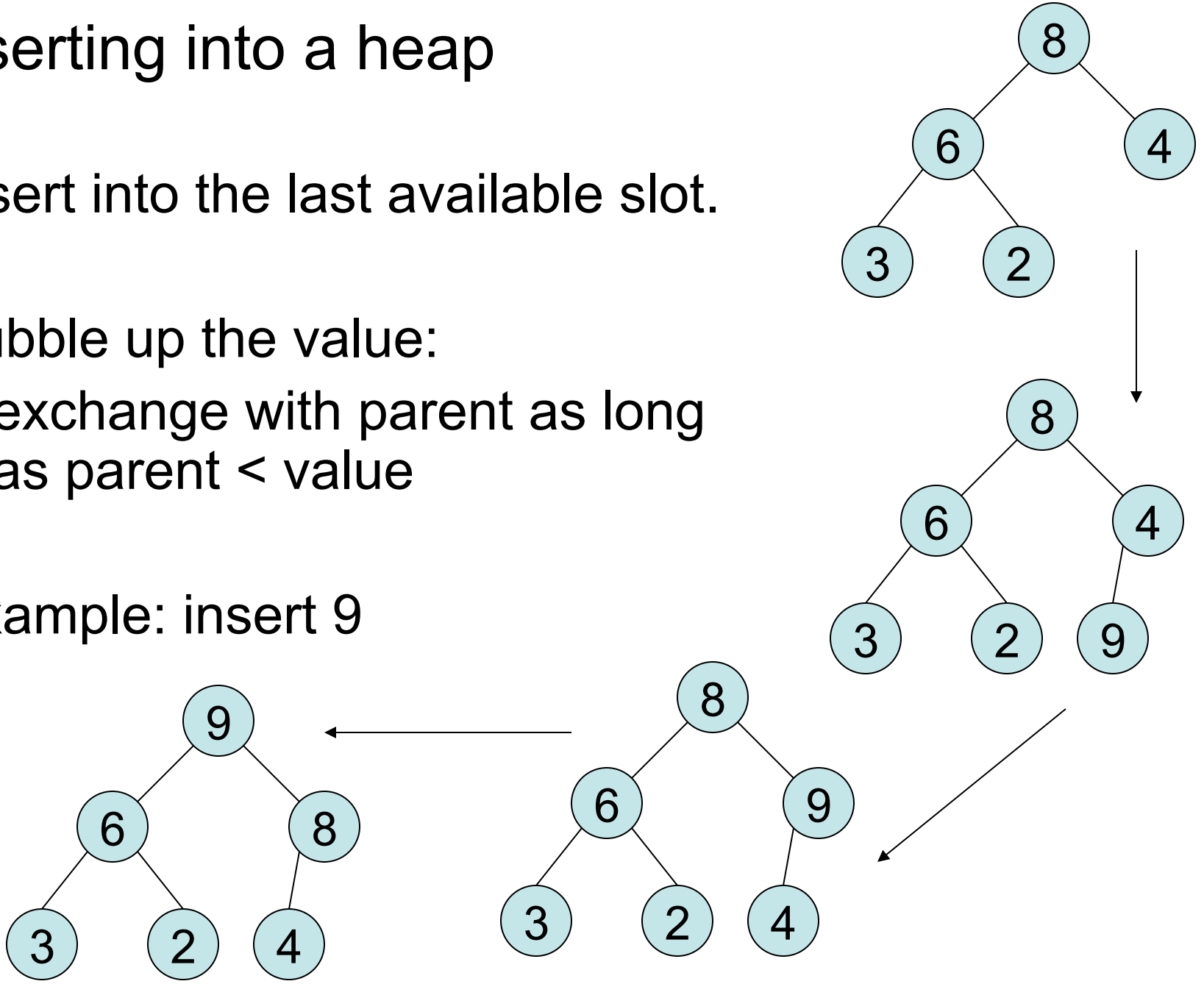
Inserting into a heap

Insert into the last available slot.

Bubble up the value:

exchange with parent as long as parent < value

Example: insert 9



Removing from a heap

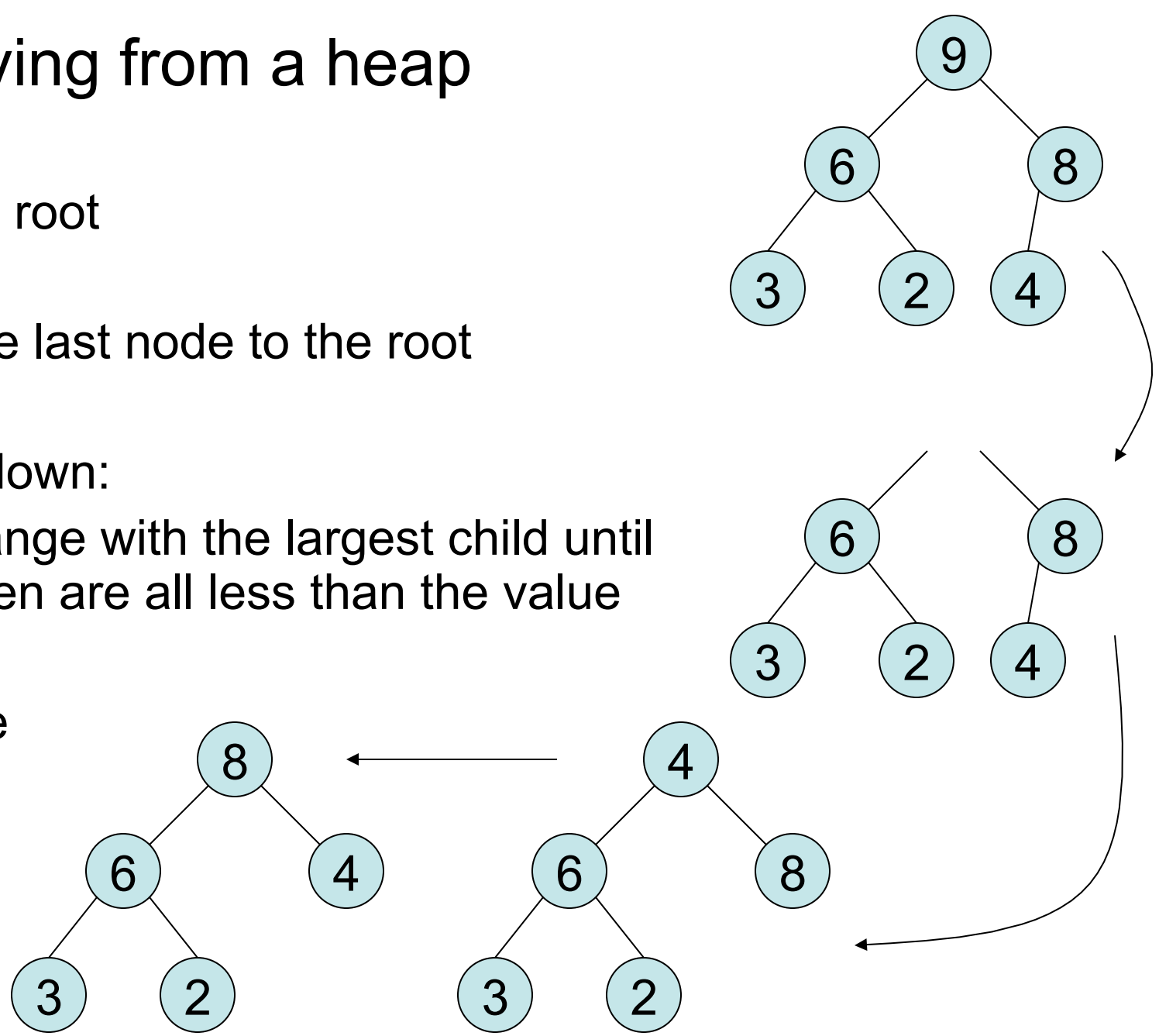
Remove root

move the last node to the root

bubble down:

exchange with the largest child until children are all less than the value

Example



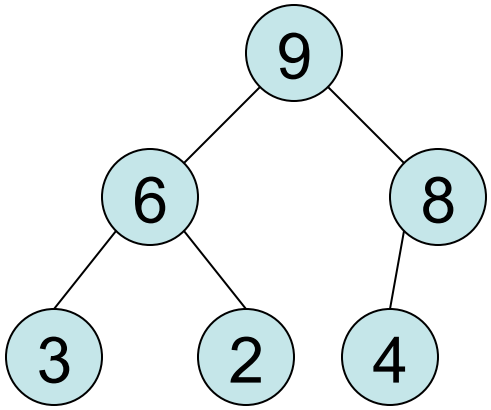
Complexity

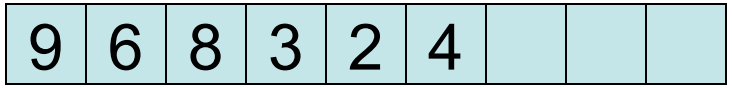
What is the complexity of heap insert?

What is the complexity of heap delete?

Implementing heaps using an array

Because a heap is a complete binary tree it can be efficiently represented as an array using 1-based indexing.





Pseudo-code for array-based heap insert

```
insert(ItemType array[], ItemType item)
   i = heapsize + 1
   j = i >> 1
  while( (j >= 1) and (array[j] < item) )
      array[i] = array[j]
      i = j
      j = j >> 1
 endwhile
array[i] = item
heapsize = heapsize + 1
```

Pseudo-code for array-based heap remove

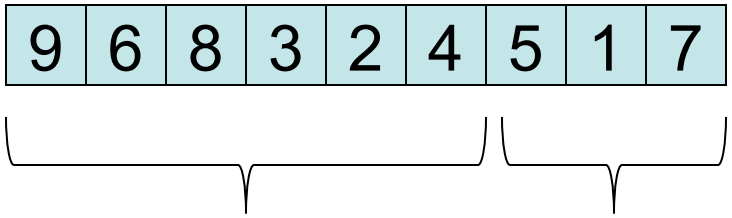
```
ItemType remove(ItemType array[])
   temp = array[1]
   array[1] = array[heapsize]
   heapsize = heapsize-1
   i = 1 and j = 2
   while( j <= heapsize)
     if( j < heapsize and array[j] < array [j+1]) j = j+1
     if( array[i] < array[j] ) swap(array[i], array[j])
     else break
     i = j
     j = j << 1
 endwhile
 return temp
```

Building an array-based heap in place

Treat the first n positions as a heap of size n.

Position n+1 is the next to insert in the heap.

Continue while n <= N



In class exercise

Given an array with the following values, show each step as you convert it to a max heap.

int a[] = {7,2,9,4,1,5};

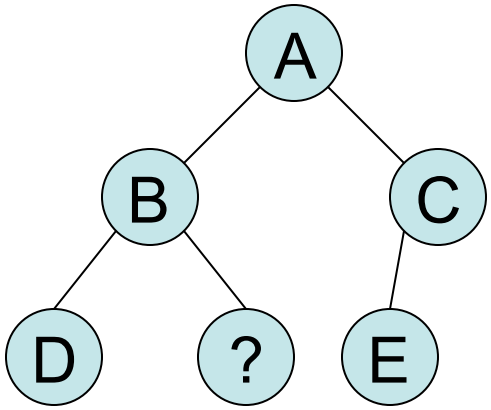
Heap Priority Queue using DynamicArrayList

See code.

Array-based incomplete binary trees

Define a special value as the missing value.

Traditionally done as a union or array of pointers, but there is an experimental std::optional type.



A	B	C	D	?	E			

Next Actions and Reminders

Read CH 6th edition pp. 459-482

Program 4 due 11/17 by 11:59 PM