

# For Friday

- Read Weiss, chapter 7, section 4
- Homework
  - Weiss, chapter 6, exercises 2-3. Do 3 only for 2a. Note that the correct answers for 2a and 2b are different.

# Programming Assignment 1

- Any questions?

# Priority Queues

- Same basic operations as a standard queue:
  - insert an item
  - delete an item
  - look at first item
  - check for empty queue
- But, order of item removal is not based on the order of item insertion (as in stacks and queues)
- Instead, each item has a **priority** associated with it

# Priority Queue ADT

- **AbstractDataType** *MaxPriorityQueue* {  
    **instances:**  
        finite collection of elements; each  
        with a priority  
    **operations:**  
        *Create()*  
        *Size()*  
        *Max()*  
        *Insert(element)*  
        *DeleteMax()*  
}

# Uses of a Priority Queue

- Operating systems
- Best first search
- Simulations
- Others?

# Implementation

- Unordered linear list
  - Insert time
  - Delete time
- Ordered linear list
  - Insert time
  - Delete time

# Min Tree

- A tree (binary or not)
- Each child has a value bigger than its parent
- Or each parent has a value smaller than any of its children (if any)
- So the smallest value in the tree is ?
- Maximum trees are simply reversed

# Heaps

- A minimum binary heap is a min tree that is also a complete binary tree
- Usually represented in an array
- Height of a complete binary tree in terms of  $N$ ?

# Heap Operations

- Insert
- DeleteMin
- DecreaseKey
- IncreaseKey
- Remove
- BuildHeap

# Sorting

- Importance of sorting
- Three basic simple sorts
  - bubble (or exchange)
  - selection
  - insertion

# Bubble Sort

- Concept is to bubble the largest to the top (or the smallest to the bottom)
- Also called exchange sort.
- Naïve vs. improved bubble sort.

# Selection Sort

- Basic concept to the find the smallest (or largest) remaining element and put it in place.

# Insertion Sort

- Basic concept:
  - Conceptually split the list to be sorted into two parts: one that is sorted and one that is not
  - Repeatedly insert the first element from the unsorted part into the sorted part.

# Performance of Sorting

- What is the performance?
- Each sort would be the best choice (of the simple sorts, at least) in certain situations—what situations?
- What's the space cost of these sorting algorithms?