Sorting

Heapsort

- **Build a heap containing the n items** to be sorted (requires O(n log n))

- Remember that any binary tree can also be implemented with an array

  (1) Nodes in tree stored in partially filled array called *data*
  (2) Root node is in *data[0]*
  (3) For node in *data[i]*, its left child is in *data[2i + 1]* and its right child is in *data[2i + 2]*
  (4) For node in *data[i]*, its parent is in *data[(i – 1)/2]*
  (5) *Used* and *capacity* of array are maintained

- **Call deleteRoot n times** (requires O(n log n))

- Therefore, heapsort requires **O(n log n)**

…What if array is already in sorted order???
…What if array is already in descending sorted order???

Suggestions for HW #9

Copy Constructor

- Constructor that has 1 parameter that is of the same type as the class
- The one parameter is usually declared with **const &**
- Called when:
    1. a class object is declared and initialized by another object of same type
    2. when a function returns a value of the class type
    3. whenever an argument of the class type is passed by value
- Not called when you set one object equal to another with =

```
HEAP::HEAP(const HEAP &source) { … }

void CEmergencyMedicalClinicDlg::OnBnClickedButtonlistpatients( )
{
if (UpdateData(true))
{
  treatmentListComboBox.ResetContent( );
  HEAP tempHeap(patientHeap);  // use copy constructor for heap
  while (! tempHeap.isEmpty( ))
  {
    …
    treatmentListComboBox.AddString(rootEntry);
    tempHeap.deleteMaxNode( );
  }
  treatmentListComboBox.SetCurSel(0);
 }
UpdateData(false);
}
```