Sorting continued

Eugeniy E. Mikhailov

The College of William & Mary

Lecture 09
Heap is a structure where parent element is larger or equal to its children.

The top most element of the heap is called root.
Heap sorting method

1. Fill the heap from the input vector elements
   - 1. take the element and place it at the bottom of the heap
   - 2. sift-up (bubble up) this element
   - 3. do the same with the next element

2. remove the root element since it is the largest

3. rearrange the heap i.e. sift-down
   - 1. take the last bottom element
   - 2. place it at the root
   - 3. check if parent is larger then children
     - 1. find the largest child element
     - 2. if the largest children is larger then parent swap them and repeat the check

4. repeat step 2 until no elements left in the heap

Heap sorting complexity $O(N \log_2 N)$
Filling (sift-up) the heap

**Step 1**
Place new element at the bottom of the heap

![Heap Diagram]

Eugeniy Mikhailov (W&M)
Practical Computing
Lecture 09
Filling (sift-up) the heap

Step 2
Check if parent is larger than child. If so swap them and repeat step 2.
Filling (sift-up) the heap

Step 2
Check if parent is larger than child. If so swap them and repeat step 2.
Filling (sift-up) the heap

Step 2
Check if parent is larger than child. If so swap them and repeat step 2.
Removing from the heap (sift-down) the heap

Step 1
Remove the root element
Removing from the heap (sift-down) the heap

Step 2
Place the last element of the heap to the root
Removing from the heap (sift-down) the heap

Step 3
Check if parent is smaller than the largest child. If so swap and repeat step 3 else go to step 1
Removing from the heap (sift-down) the heap

Step 3
Check if parent is smaller than the largest child. If so swap and repeat step 3 else go to step 1
Removing from the heap (sift-down) the heap

Sequence repeats

**Step 1**
Remove the root element
Heap nodes are numbered consequently these numbers represent the node position in the vector.

notice that parent and children have very simple relationship

- if parent node index is \( i \)
  - child 1 index is \( 2i \)
  - child 2 index is \( 2i + 1 \)
- if we know child index \( i \) then
  - parent index is \( \text{floor}(i/2) \)