CS367 Announcements
Thurs, July 24th, 2013

- H6 due Mon 6pm
- P3 due Wed July, 31st 11:59pm

Last Time

- Priority Queues
- Heaps

Today

- Heaps (cont.)

Heaps

Heap is a complete binary tree with the ordering constraint that for each node N, the value in N is larger than (or equal to) any of the values in N's subtrees

A heap as an array of values:

- root is at A[1]
- for each "node" A[i]
  - parent is at A[i/2]
  - left child is at A[2i]
  - right child is at A[2i+1]

Example: given the following heap (represented as an array):

	64	52	35	46	17	15	34	12	23	14	

What does it look like as a binary tree?

Inserting into a Heap

Practice:

initial heap:

	64	52	35	46	17	15	34	12	23	14		

Show the heap after adding 36 and 57:

Inserting into a Heap (cont.)

Heap class data members:

```
private Comparable[] items;
private int nextLoc;
```

Pseudo-code:

```
private void insert(Comparable data) {
```

Removing from a Heap

Practice:

Heap after adding 36 and 57:

What will the heap look like after doing 2 remove operations?

Heap Complexity

Is a heap *value*-oriented or *position*-oriented?

Complexities:

`isEmpty`

`getMax`

`insert`

`removeMax`

Using a Heap for Sorting: