Announcement: Midterm Exam: Oct. 29th, in class

Last time: Heap
Today: Heapsort and Binary Search Trees (BST)

I Heapsort
1. heap

2. Heapsort(A)

Build_heap(A) \hspace{1cm} T=O(n)
For i←length(A): -1: 2
   Swap(A[1],A[i])
   heapsize←heapsize-1
   heapify(A, 1)
end \hspace{1cm} T<O(nlogn)

for example:
A: 5 13 2 25 7 17 20 8 4
Build a heap, the steps are as follow:

T=O(n)
To sort, the steps are as follow:

Time=O(nlogn)

II Binary Search Trees
Goal: to support insert, delete, member, minimum, maximum fast

BST property:

x.left
x.right
x.parent
x.key

1. Print all elements sorted

in_order_tree_walk(x)                        %here x is the pointer%
if x ≠ NIL
    in_order_tree_walk(x.left)
    print (x.key)
    in_order_tree_walk(x.right)
end

2. member(x, k)
if x=NIL ∨ k=x.key
    return x
end
if k<x.key
    return member(x.left, k)
else
    return member(x.right, k)
end
3. Minimum
while x.left ≠ NIL
    x ← x.left
end
return x

4. Maximum(x)
While x.right ≠ NIL
    x ← x.right
end
return x

5. Insert(IDEA)
   • Always insert as leaf
   • Follow BST Property and go down the right branch
Example:

```
Insert 10
12
  7
  15
  10
  14
  25

Insert 11
12
  7
  15
  10
  14
  25
  11

Insert 11.5
12
  7
  15
  10
  14
  25
  11
  11.5
```

6. Find the next element
Tree_successor(x)
If x.right ≠ NIL
    Return minimum(x.right)
Else
    y ← x.parent
    while y ≠ NIL ∧ x = y.right
        x ← y
        y ← y.parent
    end
    return y
end
Here if \( x=5 \)
\( x.\text{right} \neq \text{NIL} \)
Then successor\((x) = 7 \)

If \( x=7 \)
\( x.\text{right} = \text{NIL} \)
\( y = x.\text{parent} = 5 \)
\( x = y = 5 \)
Then the successor = \( y.\text{parent} = 12 \)

7. delete\((y)\)

If \( y.\text{left} = \text{NIL} \)
If \( y.\text{left} \neq \text{NIL} \), find \( y \)'s successor

Lamma: let \( T \) be a BST of depth \( d \) then, any of minimum, maximum, insert, delete, member operations can be executed in \( O(d) \) steps

BST with \( n \) nodes has \( d \) bounded by \( \log n \leq d \leq n - 1 \)