Late Assignment submission  
Upt. 2 days : lose 25%  
" 4 day : lose 50%  
No submission beyond 4 days  

From next submission, hand in typeset solns (hardcopy not email) preferably in latex since it is easier to typeset math formulas.

Heap data structure

Priority queue: Given a set of elements, we want to support the following operations efficiently:  
1. Find min element of S  
2. Extract-min (find and delete)  
3. insert/delete elements in/from S  
   $O(\log |S|)$  
   $|S| = n$
Making a heap out of $n$ elements takes $O(n)$ time.

Compare with binary search trees.

→ Can we search in heaps?

Yes, but...

Given two heaps $H_1$ and $H_2$, can we construct $H = H_1 \cup H_2$?

Assume $|H_1| \leq |H_2|$

Then insert elements from $H_2$ into $H_1$

$H_1 \rightarrow O\left(\log |H_1| \cdot |H_2|\right)$

For $|H_1| \sim |H_2| \rightarrow O(n \log n)$

Goal: Construct a data structure to support unions (including the basic priority queue operations) in $O(\log n)$. 
Construct $B_i$ using $2B_{i-1}$ where

$B_{i-1}$

Make the root of one of the $B_{i-1}$ the leftmost child of the root of the other $B_{i-1}$

$B_1$  $O$  $B_2$

$B_3$

The family of trees is called Binomial trees.
Claim: (1) $B_i$ has $2^i$ nodes

(2) $B_i$ has depth $i$

(3) At depth $j$ from root, $B_i$ has $\binom{i}{j}$ nodes

$$\binom{n}{i} = \binom{n-1}{i} + \binom{n-1}{i-1}$$

Use this to prove by induction.

(4) Maxm no. of children at any node (root) is $i$.

Binomial Heaps

are collections of ordered Binomial trees whose nodes satisfy the heap property (min heaps)

$B_5$, $B_5$, $B_3$, $B_1$

$32 + 32 + 8 + 1 = 73$ nodes
Store the roots of the binomial trees in a list

\[
\begin{align*}
B_0 &\rightarrow B_1 \\
\triangle &\rightarrow \triangle \\
\circ &\rightarrow \circ
\end{align*}
\]

To report/find the min, we traverse the root list and identify the smallest.

\[O(\# \text{ binomial trees})\]

13 : 1101 = \(2^3 + 2^2 + 2^0\)

\[\Rightarrow \text{At most } \log_2 n \text{ trees } B_3, B_2, B_0\]

Moreover it is unique

Finding min in \(O(\log_2 n)\) time

\[
\begin{align*}
B_{H_1} &\rightarrow 0 \\
\rightarrow \circ &\rightarrow \circ \\
\triangle &\rightarrow \circ
\end{align*}
\]

\[
\begin{align*}
B_{H_2} &\rightarrow 0 \\
\rightarrow \circ &\rightarrow \circ \\
\triangle &\rightarrow \circ
\end{align*}
\]

\[
\begin{align*}
\vec{0} &\rightarrow 2 \\
\vec{0} &\rightarrow 6 \\
\circ &\rightarrow 3 \\
\circ &\rightarrow 5 \\
\circ &\rightarrow 9 \\
\circ &\rightarrow 0
\end{align*}
\]

\[
\begin{align*}
\vec{0} &\rightarrow \Delta_1 \\
\Delta_0 &\rightarrow \Delta_1 \\
\Delta_3 &\rightarrow \Delta_2 \\
\Delta_6 &\rightarrow \Delta_2
\end{align*}
\]

- 101101

- 101111

- 0101000
Union of two binomial heaps can be done in time
\[ O(\log(n_1) + \log(n_2)) \]
i.e. \[ O(\log n) \]

\[ n = n_1 + n_2 \]