CSL 356 Lecture 9 Aug 13

Late Assignment submission

Upto 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 formulae.

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

$O(\log |S|)$ $\quad |S| = n$

Making a Heap out of n elements take $O(n)$ time

Compare with binary search-trees

→ Can we search in heaps?

Yes but ...

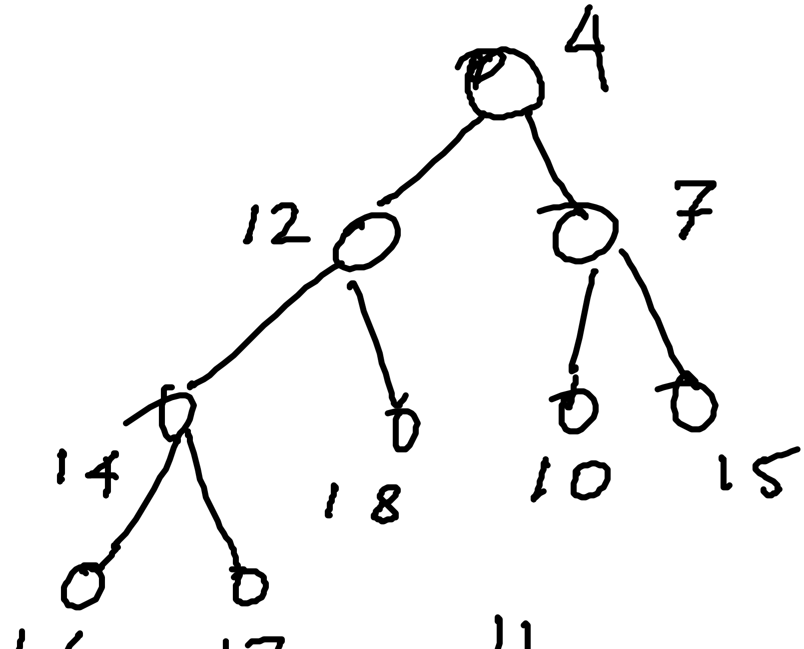
Given two heaps H_1 and H_2 can we construct $H = H_1 \cup H_2$

Assume $|H_1| \leq |H_2|$

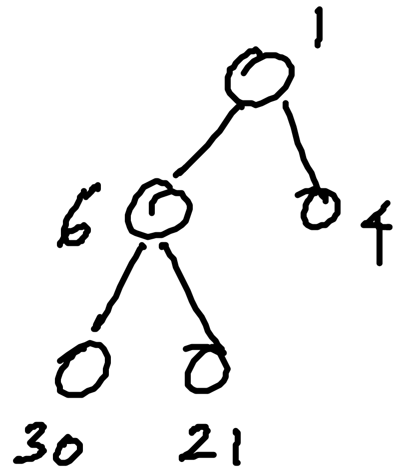
Then insert elements from H_2 into H_1 $\Rightarrow$ $O(\log|H_1| \cdot |H_2|)$

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

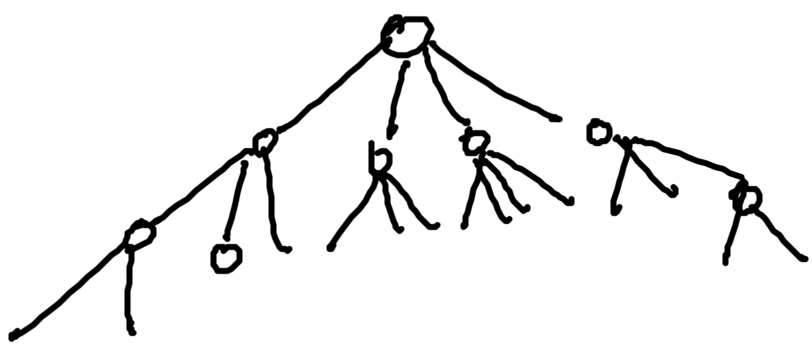
Goal: Construct a data structure to support unions (including the basic priority queue operations) in $O(\log n)$



H_1



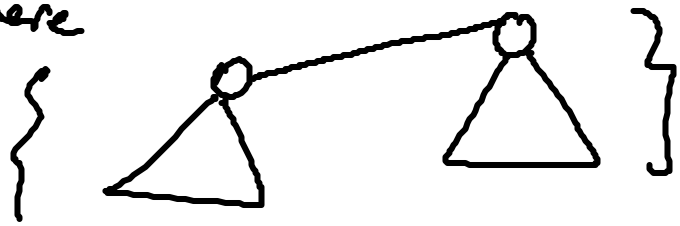
H_2



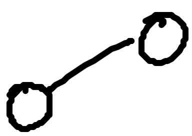
B_0

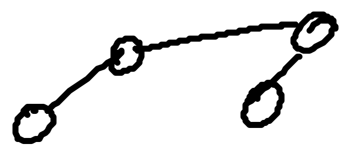
Construct B_i using 2 B_{i-1}

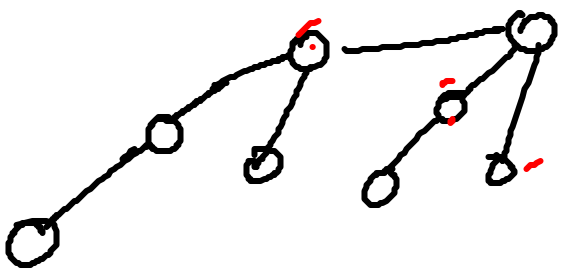
where

B_{i-1}  B_{i-1}

Make the root of one of the B_{i-1} the <u>left most</u> child of the root of the other B_{i-1}

B_1 

B_2 

B_3 

level 0	:	1
level 1	:	3
level 2		3
level 3		1

$i = 3$

This family of trees is called Binomial trees

<u>Claim</u> : (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

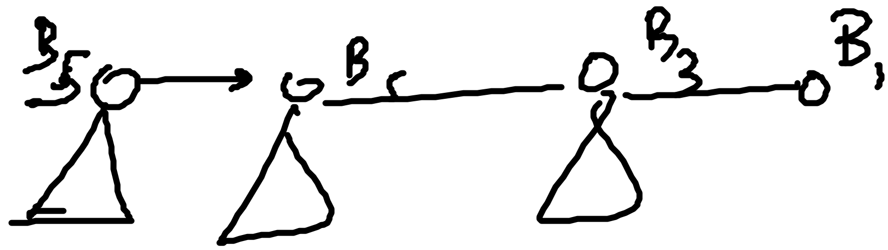
(4) Maxm no. of children at any node (root) is i.

<u>Binomial Heaps</u>

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

$$\begin{array}{ccccccc} B_5 & & B_5 & & B_3 & & B_1 \\ 32 & + & 32 & + & 8 & + & 1 \end{array} = 73 \text{ nodes}$$

Store the roots of the Binomial trees in a list



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

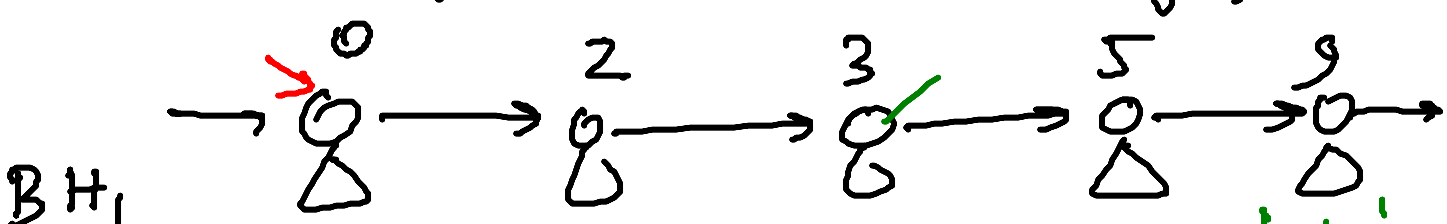
$O(\#\text{ binomial trees})$

$13 : 1101 = 2^3 + 2^2 + 2^0$

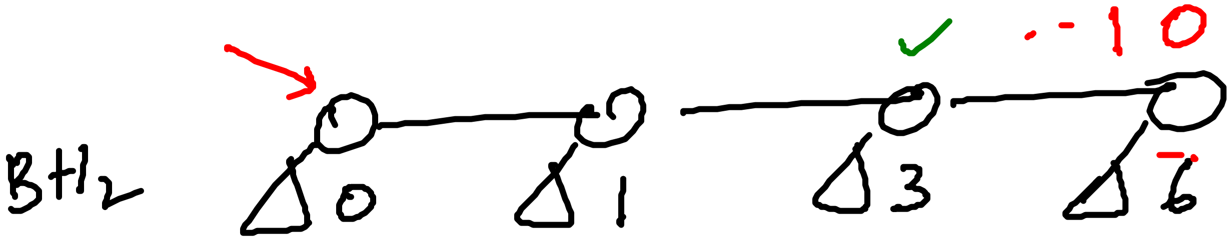
$B_3 \quad B_2 \quad B_0$

$\Rightarrow$ At most $\underline{\log n}$ trees
Moreover it is unique

Finding min is $O(\log n)$ time



$\cdots 101101$



1011

010.1000

Union of two binomial heaps can be done in time

$$O(\log(n_1) + \log(n_2))$$

i.e. $O(\log n)$

$n = n_1 + n_2$