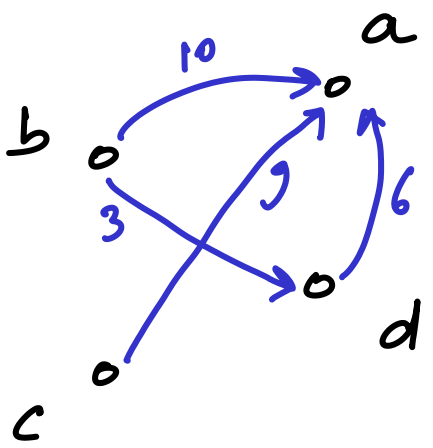
COL 702 Lecture 4 Aug 3

Shortest path computation using Matrix representation (Adjacency matrix)



$A =$

	a	b	c	d
a	0	∞	∞	∞
b	10	0	∞	3
c	9	∞	0	∞
d	6	∞	∞	0

$\infty + \square = \infty$

$A \otimes A = B : B_{ij} = \min_k \{A_{ik} + A_{kj}\}$

(matrix $A \times A$ $= \sum_{k=1}^{n} a_{ik} \times a_{kj}$)

0	∞	∞	∞
10	0	∞	3
9	∞	0	∞
6	∞	∞	0

0	∞	∞	∞
10	0	∞	3
9	∞	0	∞
6	∞	∞	0

$=$

	a	b	c	d
a	0	∞	∞	∞
b	9	0	∞	3
c	9	∞	0	∞
d	6	∞	∞	0

Claim: $A^i = A \oplus A \otimes \cdots A$ (i times) gives us shortest paths with at most i edges

$\Rightarrow$ A^{n-1} gives us APSP

Dijkstra's algorithm uses Heap

- Min of all labels : Extract min
- Change (decrease) labels ✓

Using Heaps : $O(\log n)$ for both operations

Heap : min, Extract min, insert, delete,
(priority queues) decrease → delete + insert

Given 2 Heaps H_1 and H_2 create
a new heap $H_1 \cup H_2$

Given 2 dictionaries, how do we combine them

Simplest scheme will be to insert
element by element from the 'smaller'
to the 'larger'.

$|H_1| : \sqrt{n}$ $\quad H_2 : n - \sqrt{n}$

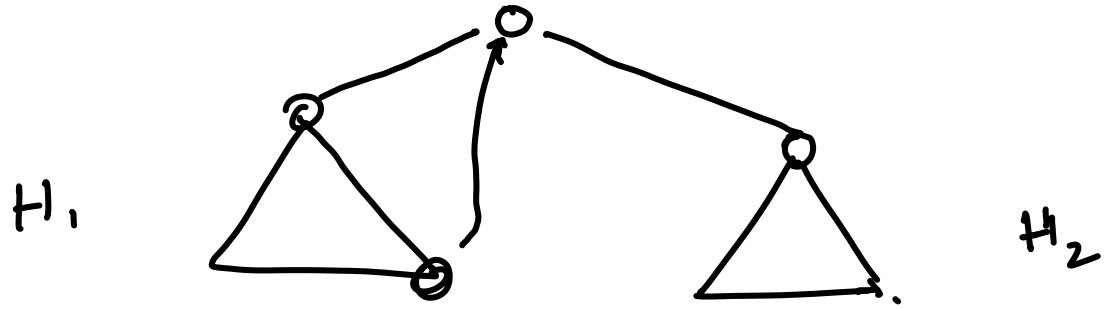
$H_1 \rightarrow H_2$ $\quad \sqrt{n} \cdot \log(n - \sqrt{n}) :$
$\leq \sqrt{n} \log n$

$H_2 \rightarrow H_1$ $\quad n \times \log(\sqrt{n})$

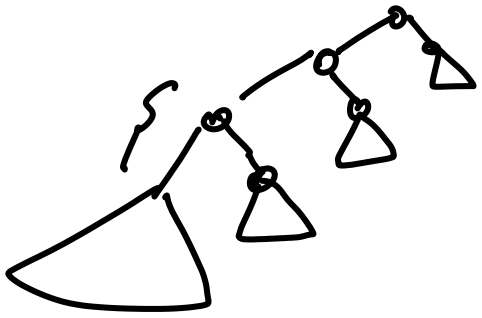
$\leq \frac{n \log n}{2} - \frac{\sqrt{n} \log n}{2}$

Notice that constructing a heap from scratch takes $O(n)$ time.

Can we combine heaps in $O(\log n)$



Take any element from the last level (delete)
create a new root node.
Then call Heapify :
Total time is $O(\log n)$

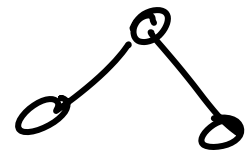


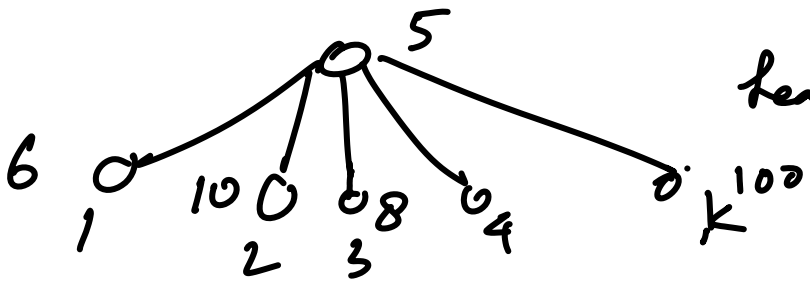
linear structure

Heap property : (min heap)
- the value at a node is no larger than children

Can we improve the performance of a heap using a k-ary structure

Normal heap: binary





heap property is maintained

Does it lead to any improvements?

$(\log_k n)$

Consider a family of Trees B_i of the following kind

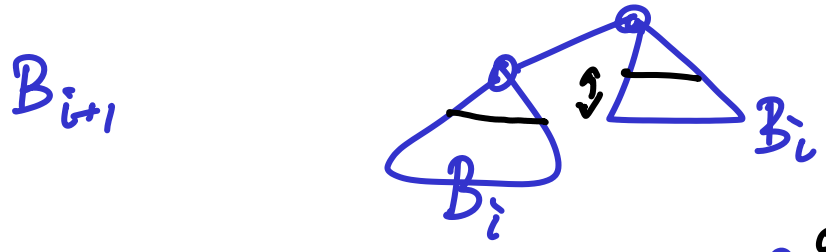
B_0 

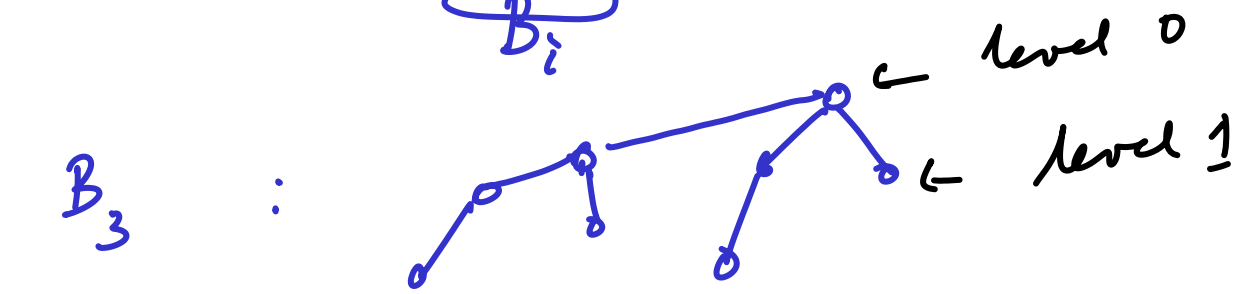
B_{i+1} (tree of $i+1$ nodes) Given two B_i trees make the root of one copy of B_i, the left child of the other copy

B_1 

B_2 

$\binom{2}{1} : 2 \quad \binom{2}{0}+1 \quad \binom{2}{2}-1$





How many nodes in B_i? 2^i

The family of trees is called Binomial Trees

Properties

1. B_i has 2^i nodes
2. The root of B_i has i children
3. The depth of B_i is i
4. B_n has ${}^nC_j$ nodes in depth j

$$\sum_j {}^nC_j = 2^n$$

B_{n-1} B_{n-1}

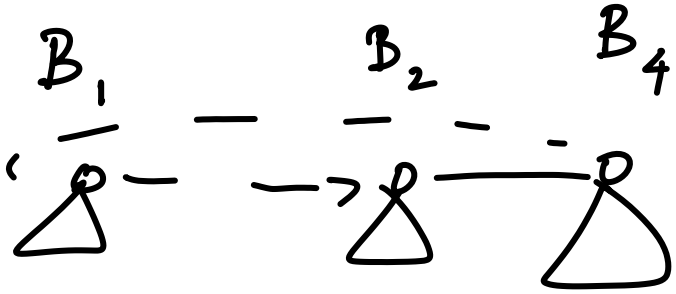
$$\binom{n-1}{j} + \binom{n-1}{j-1} = \binom{n}{j}$$

level j of one copy

level $j-1$ of the other

Binomial Heap

is a collection heap ordered Binomial Trees such that there no more than one B_j for each j and the roots are linked in a list



B_0 ~~B_1~~ ~~B_2~~ B_4

- Where is the min? One of the root nodes

Cost: no. of trees in the Heap

- Extract min: Do extract min on the tree corresponding to min

For a heap of n values, how many Binomial trees are needed?

n is not a power of 2

Binary representation of n

$n = 11$ $\quad 8 + 2 + 1$ $\quad B_3\ B_1\ B_0$