

CSE 326: Data Structures

d-Heaps

Brian Curless
Spring 2008

1

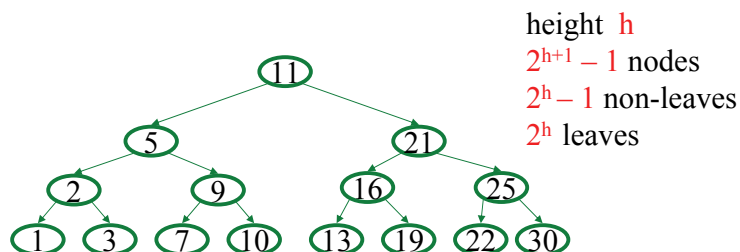
Announcements (4/9/08)

- Project #1 due today
- Homework due Friday at start of class
- Project #2a assigned Friday
 - You may work with a partner if you choose.
 - If you do so, find a partner by Friday.

2

(Aside)

A Perfect binary tree – A binary tree with all leaf nodes at the same depth. All internal nodes have 2 children.



3

Facts about Heaps

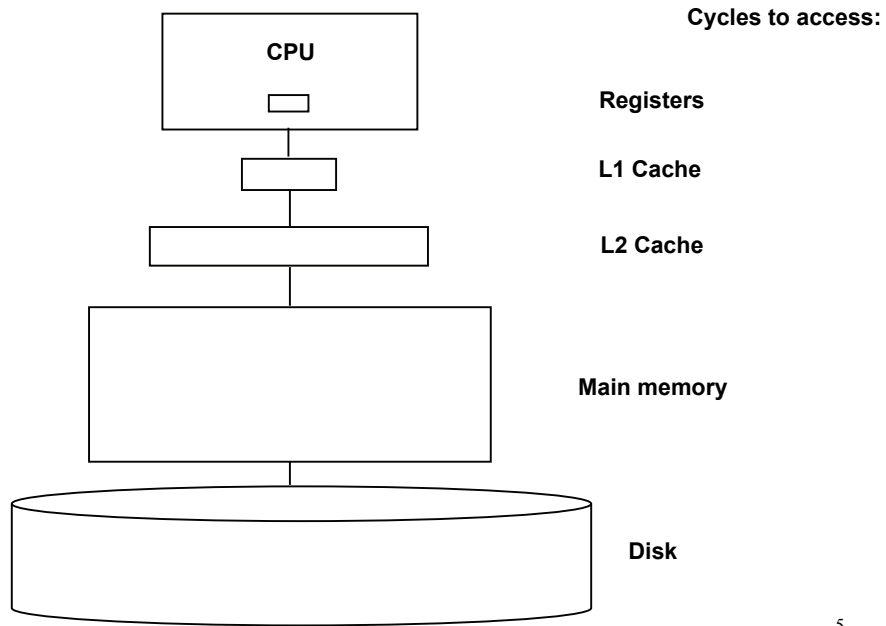
Observations:

- Finding a child/parent index is a multiply/divide by two
- Operations jump widely through the heap
- Each percolate step looks at only two new nodes
- Inserts are at least as common as deleteMins

Realities:

- Division/multiplication by powers of two are equally fast
- Hopping through the array, looking at only two new pieces of data: bad for cache!
- With huge data sets, disk accesses dominate

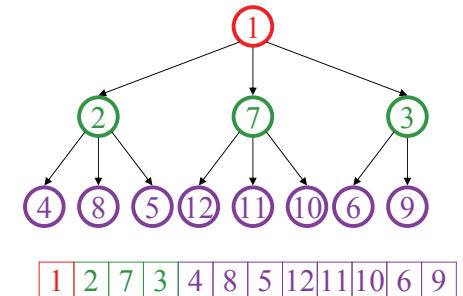
4



5

A Solution: d -Heaps

- Each node has d children
- Still representable by array
- Good choices for d :
 - (choose a power of two for efficiency)
 - fit one set of children in a cache line
 - fit one set of children on a memory page/disk block



6

Operations on d -Heap

- Insert : runtime =
- deleteMin: runtime =

Does this help insert or deleteMin more?