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

A Solution: d-Heaps

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

How does height compare to bin heap? (less)

This example has height 2, vs. 3 for bin
Operations on $d$-Heap

• Insert: runtime = depth of tree decreases, $O(\log_d n)$ worst

• deleteMin: runtime = percolateDown requires comparison to find min, $O(d \log_d n)$, worst/ave

Does this help insert or deleteMin more?