# Disks

- – Seek time: moving read-write head (9 ms)
  – Rotational latency: data moving into position under r/w head (2 ms)
- Access time: 10 ms
- Blocks: fixed size chunk of data: 512 bytes - 16 kb

## File Organization

- Storing a relation on disk
- Many options:
  – Structured storage
  – Unstructured ("heap") storage
- Ordered Storage
  – Within blocks order primary key
  – Blocks recorded in PK order
  – Reads: lookup by binary search; $O(log_2(n))$
  – Inserts: expensive; easier if maintain some empty space
  – Deletions: similar
- Analysis
  –
- B-trees
  – Balanced search tree
  – 'large' number of entries per node $\equiv$ large number of children of each node
  – Def of B-tree of order m
    1. Every node has at most m children
    2. Every non-leaf node (except root) has at least $\frac{m}{2}$ (ceiling) children
    3. The root has at least 2 children (unless a leaf)
    4. A non-leaf node with k children has k-1 leaves
    5. All leaves appear on some level
  – B-Tree of Order 4 Example
    *

# 1 11/3/15

- Query in SQL flowchart
  - Query in SQL -¿
  - parse, validator -¿
  - internal representation (query tree/relational algebra
  - query
- Select (simple):
  1. linear search
  2. binary search (if records are ordered by attribute in search condtion
  3. index scan
     - Primary key is secondary
     - Equality is range
  4. "exotic" options
     - bitmap indexes
     - hash index
- Select: conjunctive (AND)
  1. linear search
  2. index on one attribute:
     - retrieve tuples matching condition on indexed attribute
     - linear search on result for other condition(s)
  3. composite index
  4. two indexed
     - separately select on each condition (index scan)
     - intersection of record pointers
- Select: disjunction (OR)
  - If any conditon cannot use an index → must use linear search