CS150A Quiz03

B+ Trees

order: max fan-out = 2d+1

Q1: Suppose that all nodes in our B+ tree have an order of 1500. What's the MAXIMUM number of records we can index with a B+ tree of height 2?

Assume our B+ trees are laid out as in lecture.

max fan-out = 2order+1 = 2×1500+1 = 3001 ; height 2

max number of records = $2d(2d+1)^h = 3000 \cdot 3001^2 = 27018003000$

AD

Q2: We want to bulk-load a B+ tree, and we increase the fill factor ↑ of this bulk load. Which of the following applies, in general?

Check all that apply.

A. The bulk loading operation is faster

B. The bulk loading operation is slower

C. We consume more disk space

D. A sequence of many consecutive record lookups is faster height ↓

E. A sequence of many consecutive insertions requires fewer disk writes

Q3: We insert the key 60 into the B+ tree in Figure A. How many I/Os (page reads and writes) does this operation take?

Assume we require zero page reads and one page write to create a new page from scratch. Also assume that we do no key redistribution. Exclude disk I/Os done to data pages. Finally, assume we have 20 pages of memory available for caching pages in memory after reading them.

1. Read [70]
2. Read [15,42]
3. Read [51 61]
4. create [60, 61]
5. delete 61 in [51, 61].
6. create [60] (push, 42)
7. dele 42 in [15, 42]
8. write [42, 70].

=> Total: 8 I/O.

why B+ ? not same in inner and leaf

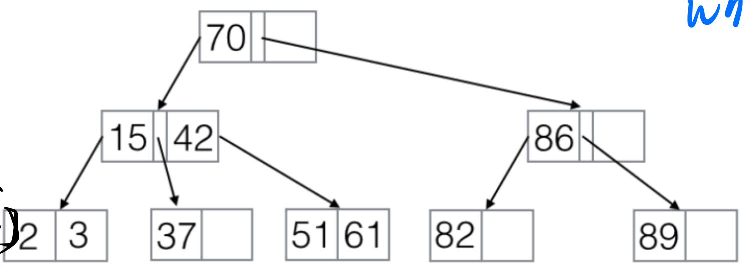
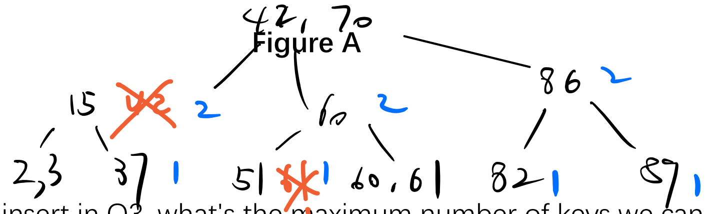


Figure A



Q4: After performing the insert in Q3, what's the maximum number of keys we can insert into the B+ tree in Figure A without splitting the ROOT?

capacity: $2d(2d+1)^h = 2\times3^2 = 18$. $d=1$

present # of entries in leaf: 8

maximum insert: $18 - 8 = 10$

or: $\underset{\text{leaf}}{1\times4} + \underset{\text{level 1}}{2\times3} = 10$