

11/19/09

32 28 53 49 75 27 65 43 21 85  
58 19 105 70 50 61  
d=2

28 32 49 53 75

49

21 27 28 32 43 → 49 53 65 75

28 49

21 27 28 32 43 49 53 65 75 85

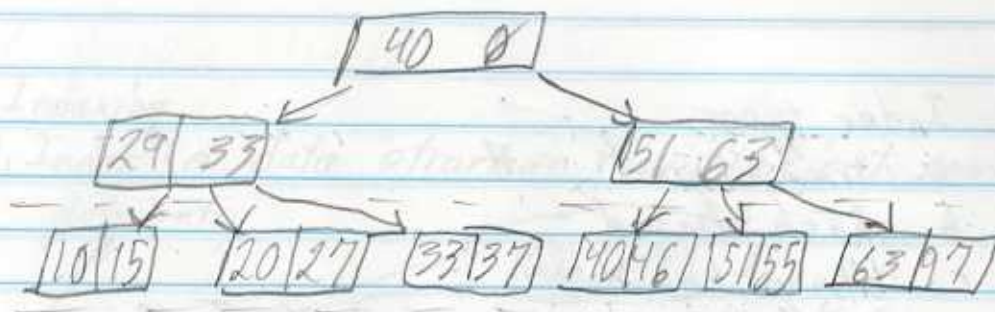
28 49 65

19 21 27 28 32 43 49 53 58 65 75 85 105

28 49 65 75 53

|    |    |    |    |     |
|----|----|----|----|-----|
| 19 | 28 | 49 | 65 | 75  |
| 21 | 32 | 50 | 70 | 85  |
| 27 | 43 | 53 |    | 105 |
|    |    | 58 |    |     |

61



10 15 20 27 33 37 40 46 51 55  
63 97 add 23, 48, 41, 42

Static tree structure, can cause performance problems

B<sup>+</sup>-Tree

1.  $d$  - ~~degree~~ ← order

each node has  $[d, 2d]$  data entries

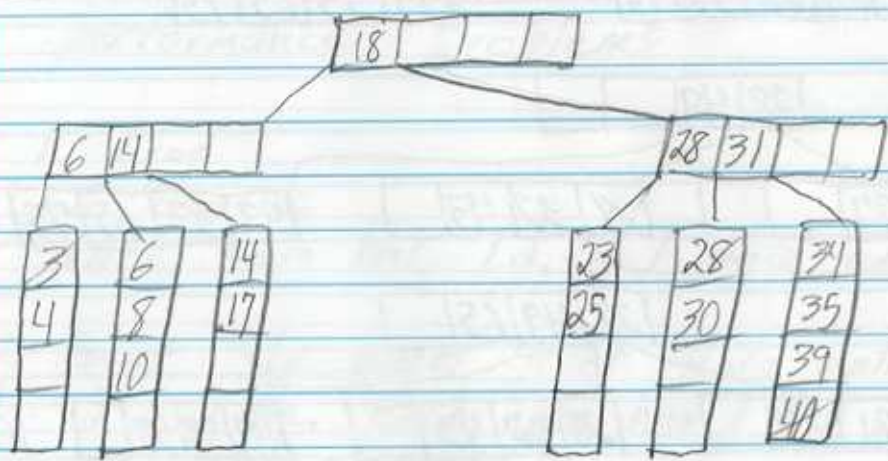
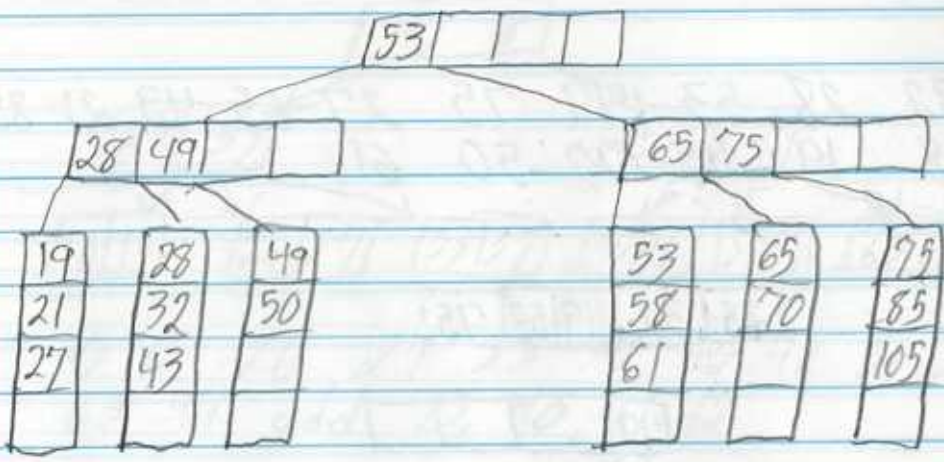
2. Fanout: the exact # of data entries in a node, each node has  $f$  key values and  $f+1$  ptrs.

3. leaf pages are chained together.

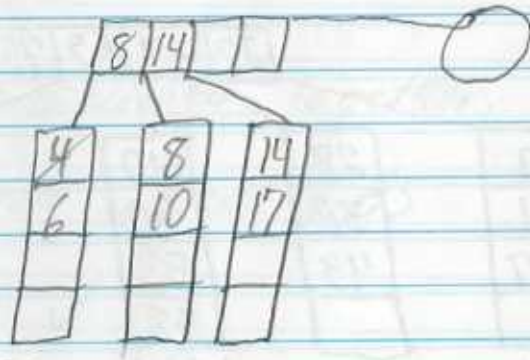
typical B<sup>+</sup>-tree in real DBS

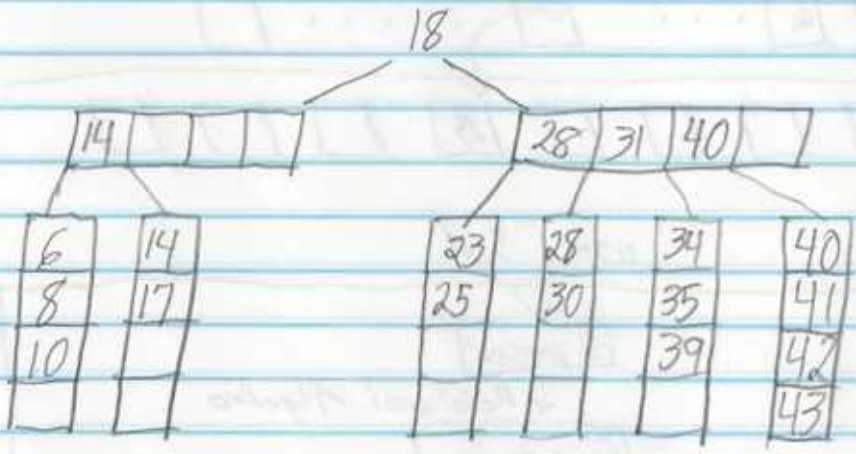
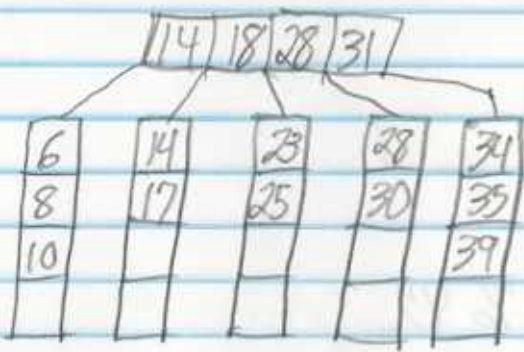
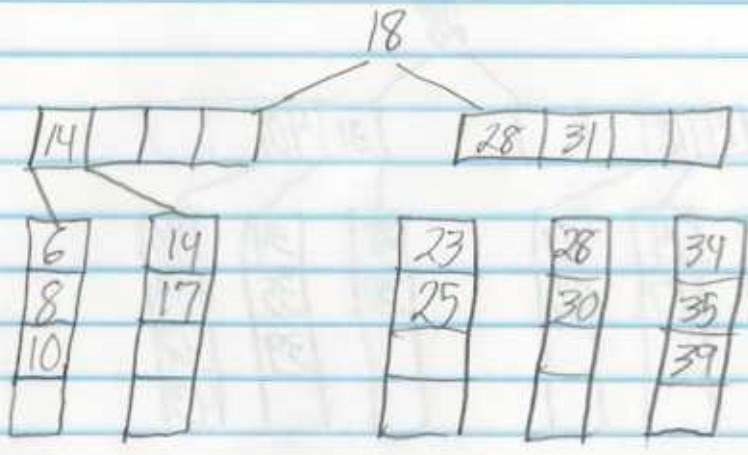
$2d: 200$

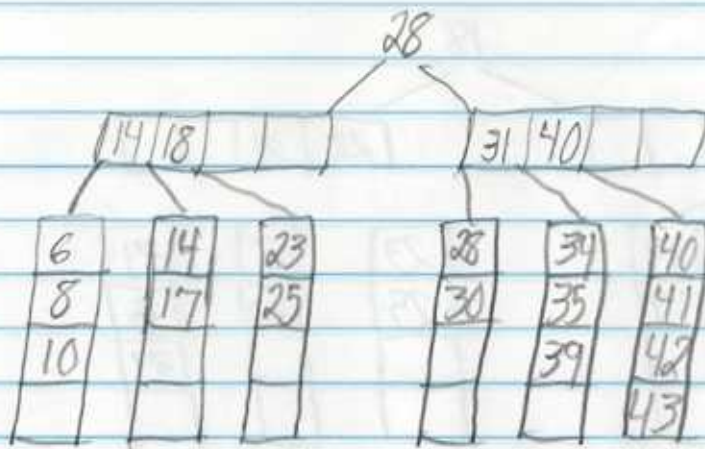
avg.  $f: 130-140$  ✓



delete 40  
 " 3  
 " 4





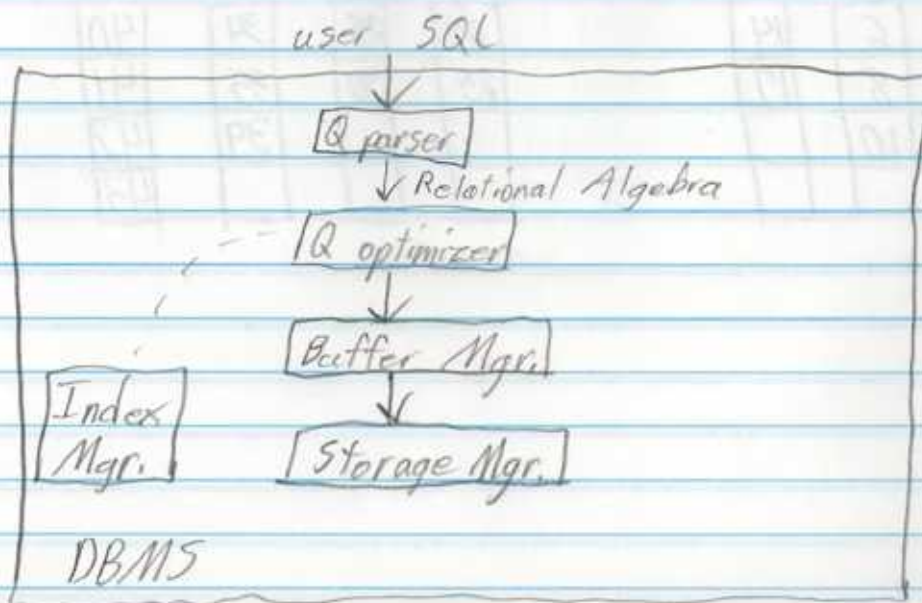
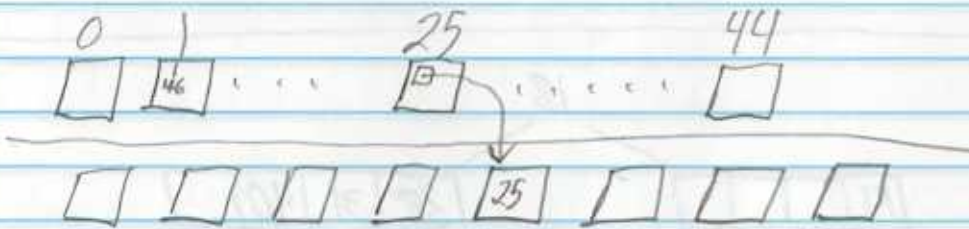


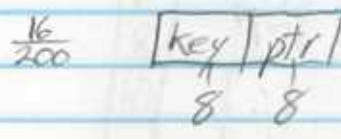
B-tree



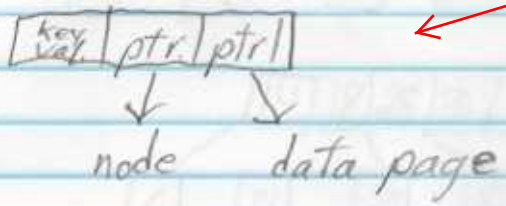
Hash indexes:

$$H-F: \lfloor \% 45$$





B-tree node



|    |    |    |
|----|----|----|
| 6  | 14 | 31 |
| 8  | 17 | 37 |
| 10 | 21 | 39 |

