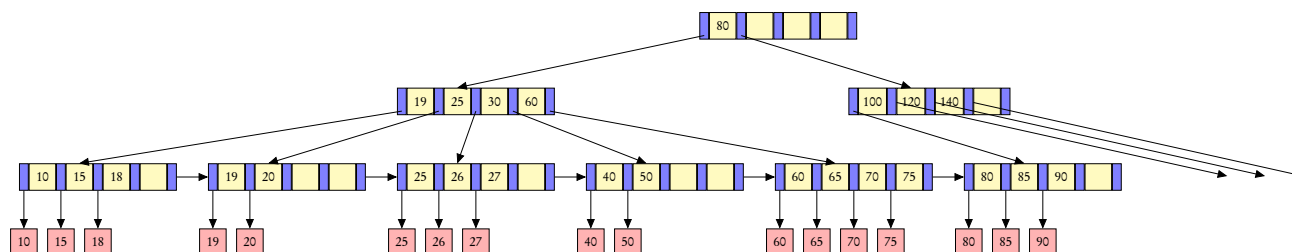


CSE 444: Homework 2

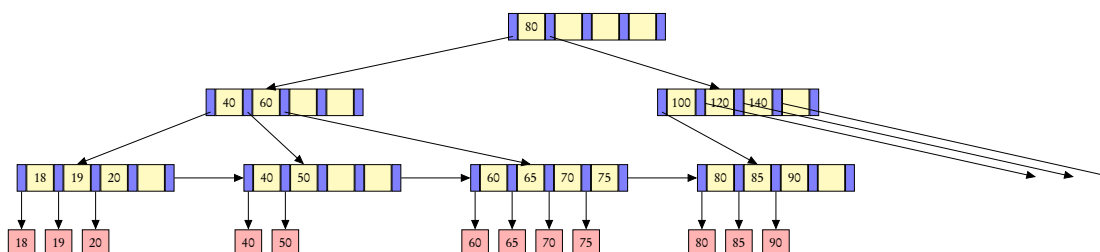
Linxing Preston Jiang Winter 2018
January 26, 2018

1 B+ Trees

1. See below



2. See below



3.
 - We need to look up 4 pages: 3 pages of index files, 1 page which contains the key 40
 - Same as above, 4 pages
 - We need to look up 3 pages to get to the key 60, and then we can keep the next page until reached 80. Then go back to the index node to find where 90 is, then read 90. So $3 + 2 + 2 = 7$ pages in total.
 - We need to look up 3 pages to get to the key 60. Because we assume the worst case of unclustered index, everything is on different pages. So we need to read 4 pages until 80 to go back to the index node, then read 80, 85, 90 on three different pages. So $3 + 4 + 1 + 3 = 11$ pages.

2 Operator algorithms

1.
 - $\text{Cost} = B(R) + T(R) \cdot B(S) = 100 + 1000 \cdot 80 = 80100$
 - $\text{Cost} = B(R) + B(R) \cdot B(S) = 100 + 100 \cdot 80 = 8100$
2. The algorithm with the least cost would be block-nested-looped. The cost is (assume S as the outer relation)

$$\text{Cost} = B(S) + \frac{B(R) \cdot B(S)}{M - 2} = 80 + \frac{100 \cdot 80}{10 - 2} = 1080$$

3. The index-based nested loop join will iterate over R , for each tuple in R , fetch the corresponding tuples from S . If we assume there is exactly one match for every tuple in S , then for every tuple in R we make 1 disk access of S . So the total cost is

$$B(R) + T(R) * 1 = B(R) + T(R) = 100 + 1000 = 1100$$

3 Multipass algorithms

1. Use partitioned hash-join. The steps are:

- (1) Partition R with a hash function h that maps the tuple into ten different pages in M (one page reserved for input buffer). Write the tuple to disk when the page is full. So in the end we have 10 partitions for R on disk. Each has length 100 blocks (assume the evenly distributing hash function).
- (2) Repeat (1) for S .
- (3) The partitions right now won't fit in memory page. So we still need another hash function h_2 to further partition the groups of R now, then write the results to disk. So now we have 100 length 10 blocks partitions for R .
- (4) Repeat (3) for S .
- (5) Read 5 blocks from R partition, 5 blocks from S partition, reserve one page for output buffer, and join until exhaustion.

Total cost: $5B(R) + 5(S) = 9000$

2. User sorted-merge-join. The steps are:

- (1) Read M blocks from R into memory and sort them, then write them out to disk. Repeat until R is exhausted. This produces 91 runs in disk of R .
- (2) Repeat (1) for S . This produces 73 runs in disk of S .
- (3) For every $M - 1$ runs in disk of R , merge them memory to produce $\left\lceil \frac{91}{11} \right\rceil = 10$ longer sorted runs.
- (4) Repeat (3) to produce one final sorted run of R written to disk.
- (5) Repeat (3) and (4) for the 73 runs in disk of S to get one final sorted run of S .
- (6) Read the two sorted runs in memory and join them.

Total cost: $7B(R) + 7B(S) = 12,600$