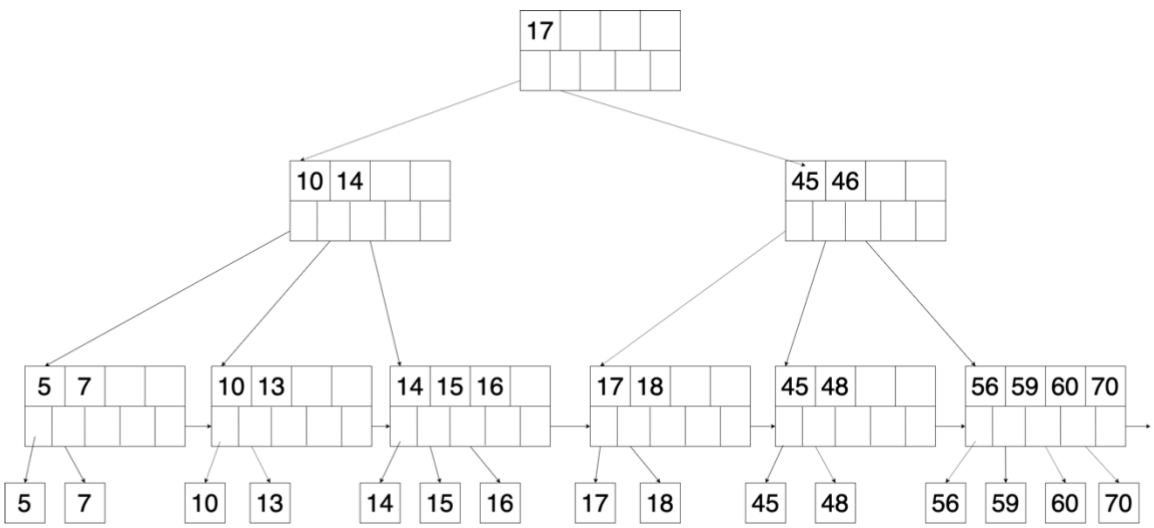
INF551
Homework4
Kaijing Zhang
USCID: 3154500265

1. (a).

-Start at the root, read the root.
-Proceed down, to the leaf.
-Find the first leaf in the range and read the leaf with node 10 and node 13.
-Continue to read next leaf with node 17 in next block and read the leaf with node 18.
-Continue to read next leaf with node 45 in next block and read the leaf with node 48.
-Since next node value is 56, which is larger than 50, stop it.

There are 4 blocks I/O's are needed for the process.

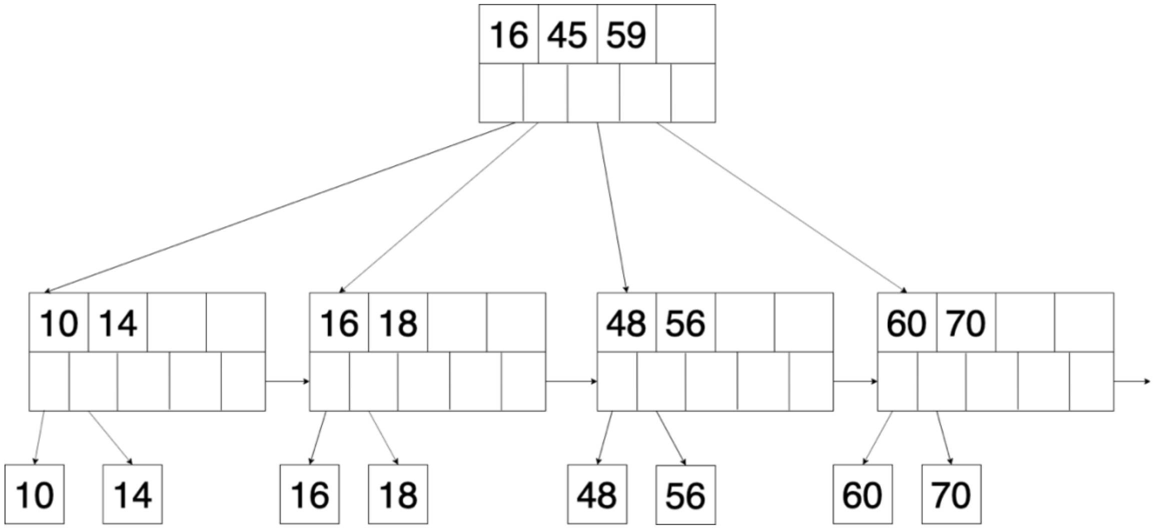
(b).



There are 3 blocks I/O's are needed for inserting 14. (read 2 blocks and write 1 block)
There are 3 blocks I/O's are needed for inserting 16. (read 2 blocks and write 1 block)
There are 7 blocks I/O's are needed for inserting 15. (read 2 blocks and write 5 block)

So total cost is 13 blocks.

(c).



There are 7 blocks I/O's are needed for deleting 5. (read 5 blocks and write 2 block)
There are 3 blocks I/O's are needed for deleting 7. (read 2 blocks and write 1 block)
There are 6 blocks I/O's are needed for deleting 13. (read 3 blocks and write 3 block)
There are 6 blocks I/O's are needed for deleting 15. (read 4 blocks and write 2 block)
There are 6 blocks I/O's are needed for deleting 17. (read 3 blocks and write 3 block)
There are 7 blocks I/O's are needed for deleting 45. (read 4 blocks and write 3 block)
There are 3 blocks I/O's are needed for deleting 59. (read 2 blocks and write 1 block)
So total cost is 38 blocks.

2. (a). for each (102-2) blocks b_r of R do

```
for each (102-2) blocks b_r of R do
    for each blocks b_s of S do
        for each tuple r in b_r do
            for each tuple s in b_s do
                if r and s join then output (r, s)
```

Read R once: cost B(R)
Outer loop runs B(R)/(M-2) times,
and each time need to read S: costs B(R)B(S)/(M-2)
Cost: B(R) + B(R)B(S)/(M-2) = 20000 + 10000 * 20000 / (102 - 2) = 2020000 blocks I/O.

(b).

```
for each (102-2) blocks b_s of S do
    for each blocks b_r of R do
        for each tuple s in b_s do
            for each tuple r in b_r do
                if s and r join then output (s, r)
```

Read S once: cost B(S)
Outer loop runs B(S)/(M-2) times,
and each time need to read R: costs B(S)B(R)/(M-2)
Cost: B(S) + B(S)B(R)/(M-2) = 10000 + 20000 * 10000 / (102 - 2) = 2010000 blocks I/O.

(c). -Sort R into B(R) / 100 = 20000 / 100 = 200 runs. Cost: 2B(R)

-Sort S into B(S) / 100 = 10000 / 100 = 100 runs. Cost: 2B(S)

-Further merge R into 200 / 100 = 2 runs. Cost: 2B(R)

-Further merge S into 100 / 100 = 1 runs. Cost: 2B(S)

-Finally join 2 runs from R and 1 run from S. Cost: B(R) + B(S)

Total Cost: 5B(R) + 5B(S) = 5 * 20000 + 5 * 10000 = 150000 blocks I/O.

(d). -Sort R into B(R) / 100 = 20000 / 100 = 200 runs. Cost: 2B(R)

-Sort S into B(S) / 100 = 10000 / 100 = 100 runs. Cost: 2B(S)

-Further merge R into 200 / 100 = 2 runs. Cost: 2B(R)

-Further merge R into 1 run. Cost: 2B(R)

-Further merge S into 100 / 100 = 1 runs. Cost: 2B(S)

-Finally join 1 run from R and 1 run from S. Cost: B(R) + B(S)

Total Cost: 7B(R) + 5B(S) = 7 * 20000 + 5 * 10000 = 190000 blocks I/O.

(e). Hash table S into (102 – 1) = 101 buckets, send all buckets to disk.

Hash table R into (102 – 1) = 101 buckets, send all buckets to disk.

Join every pair of 101 buckets from table S and 101 buckets from table R.

Cost: 3B(R) + 3B(S) = 3 * 20000 + 3 * 10000 = 90,000 blocks I/O.

(f). for each (102-2) blocks b_r of R do

    for each tuple r_r of b_r do

        fetch corresponding tuple r_s of S

Cost: B(R) + T(R)B(S) / V(S, a) = 20000 + 200000 * 10000 / 100 = 20020000 blocks I/O

Partitioned-hash join algorithm is most efficient.