**Homework #4**

**1.**

**a.**

Read the root node, read the kid node with ‘10’, read the kid node with ‘17’, read the kid node with ‘45’, the next leaf’s value is larger than 50, so stop.

**Total cost**: 4 I/O.

**b.**

Insert 14: read 2 blocks, write 1 block.

Insert 16: read 2 blocks, write 1 block.

Insert 15: read 2 blocks, write 5 blocks.

**Total cost**: 13 I/O.

**c.**

**Total cost**: 38 I/O.

**2.**

**a.**

Read R once: cost B(R) = 20,000

Outer loop runs B(R)/(M-2) times, and each time need to read S, costs B(R)B(S)/(M-2) = 2,000,000

**Total cost**: B(R) + B(R)B(S)/(M-2) = 2,020,000 blocks I/O

**b.**

Read S once: cost B(S) = 10,000

Outer loop runs B(S)/(M-2) times, and each time need to read R, costs B(S)B(R)/(M-2) = 2,000,000

**Total cost**: B(S) + B(S)B(R)/(M-2) = 2,010,000 blocks I/O

**c.**

Sort R into B(R)/M = 200 runs, cost 2B(R)

Sort S into B(S)/M = 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 merge, cost B(R) + B(S)

**Total cost**: 5B(R) + 5B(S) = 150,000 blocks I/O

**d.**

Sort R into B(R)/M = 200 runs, cost 2B(R)

Sort S into B(S)/M = 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)

Further merge R into 1 run, cost 2B(R)

Finally merge, cost B(R) + B(S)

**Total cost:** 7B(R) + 5B(S) = 190,000 blocks I/O

**e.**

Hash R into M-1=100 buckets, send all buckets to disk, cost 2B(R).

Hash S into M-1=100 buckets, send all buckets to disk, cost 2B(S).

Join every pair of corresponding buckets. Cost B(R)+ B(S).

**Total cost:** 3B(R)+ 3B(S) = 90,000 blocks I/O

**f.**

Read the table R, cost B(R).

Retrieve clustered index, average cost: B(S)/V(S,a) = 100.

Iterate over R, for each tuple, fetch tuple(s) from S.

**Total cost:** B(R) + T(R)B(S)/V(S,a) = 20,000 + 200,000\*10,000/100 = 20,020,000 blocks I/O

**Partitioned-hash join is most efficient.**