1)

a) Rblocks + [Rblocks/ B-2] \* Sblocks

R join S 100 + [100/25] \* 50 = 300 I/Os

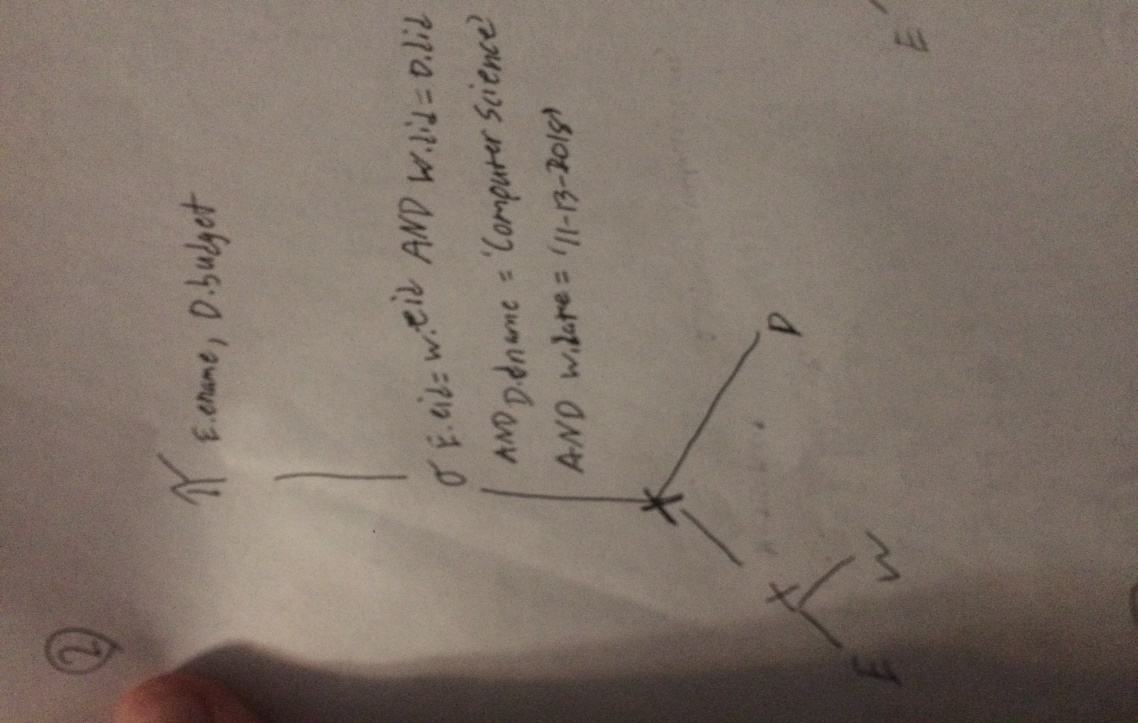
S join R 50 + [50/25] \* 100 = 250 I/Os

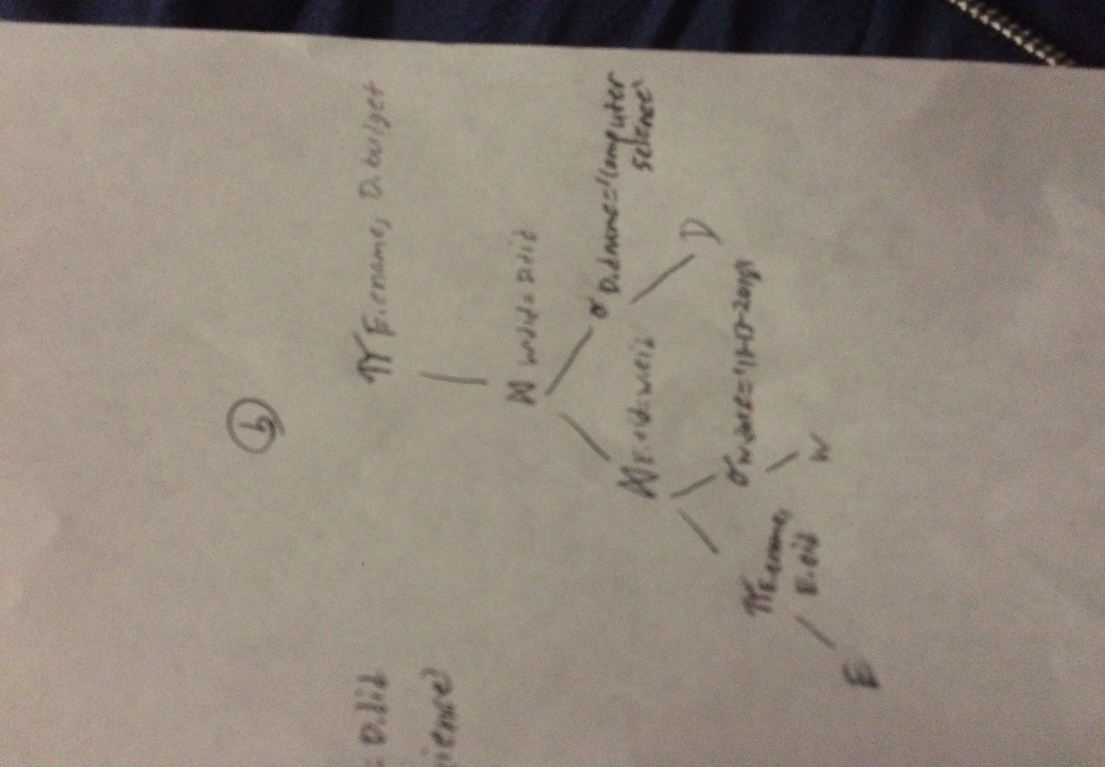
b) Rblocks + Rtuples \* (Rlevel + average-match)

R join S 100 + 4000 \* (2 + 0) = 8100 I/Os

S join R 50 + 2000 \* (1 + 2) = 6050 I/Os

2)





3)

a)

h1 h0 primary page overflow page

000 00

9

25

9

5

001 01

010 10

34

30

18

14

2

011 11

35

31

3

7

100 00

4

36

12

101 01

b)

h1 h0 primary page overflow page

44

000 00

9

25

9

001 01

010 10

34

30

18

14

2

011 11

11

35

31

3

7

100 00

4

36

12

101 01

5

LAB01

1. The second query-tree, while it certainly seems more complicated, is actually doing everything that the first is doing. However it is doing so explicitly. Creating a new table by selecting everything from dept\_manager and employees; and then checking the where condition, whereas the first tree creates the join from both tables as they exist and then checks the condition.
2. First a buffer sort and a merge join handle tables dept\_emp and dept\_manager respectively. Then the selection condition limits the available choices and the projection, which is the SELECT condition outputs the results.
3. Two index scans handle tables dept\_emp and dept\_manager. Then a hash join connects the two. A projection, which is the SELECT condition outputs the results.