# CS 290: Data Engineering Assignment 3

Cody Lieu
cal53

1) a) Figure 1(a) can be rewritten as: $\pi_{R.A}\left[(R(A) \bowtie S(A)) \bowtie T(A)\right]$

Figure 1(b) can be rewritten as: $\pi_{R.A}\left[(T(A) \bowtie S(A)) \bowtie R(A)\right]$

Since Natural Joins are commutative and associative, these are equivalent RA queries and produce the same result.

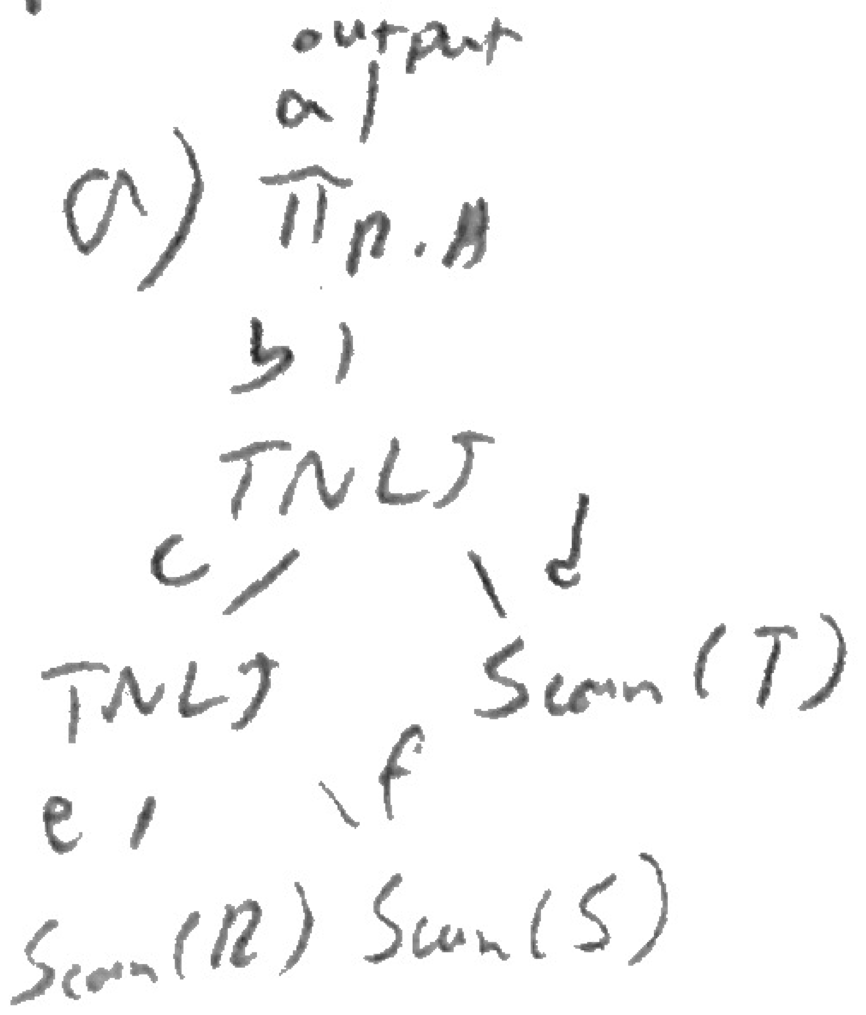
b) No, they are not equivalent. For them to produce the same results, S(A) and T(A) must not have any duplicate A values.

c) No, they are not equivalent. For them to produce the same results, T(A) must not have any duplicate A values.

d) Yes, these logical plans are equivalent.

2) I first computed 2 based off of correspondence with Junghoon on Piazza. After more clarifying questions were asked and seeing getnext.txt, I decided to redo it, but have attached my original answers to the back.

a)




Edge e: $6+1=7$

Edge f: $6(2+1)=18 \rightarrow \{(1,1),(1,1),(2,2),(2,2)\}$

Edge c: $4+1=5$

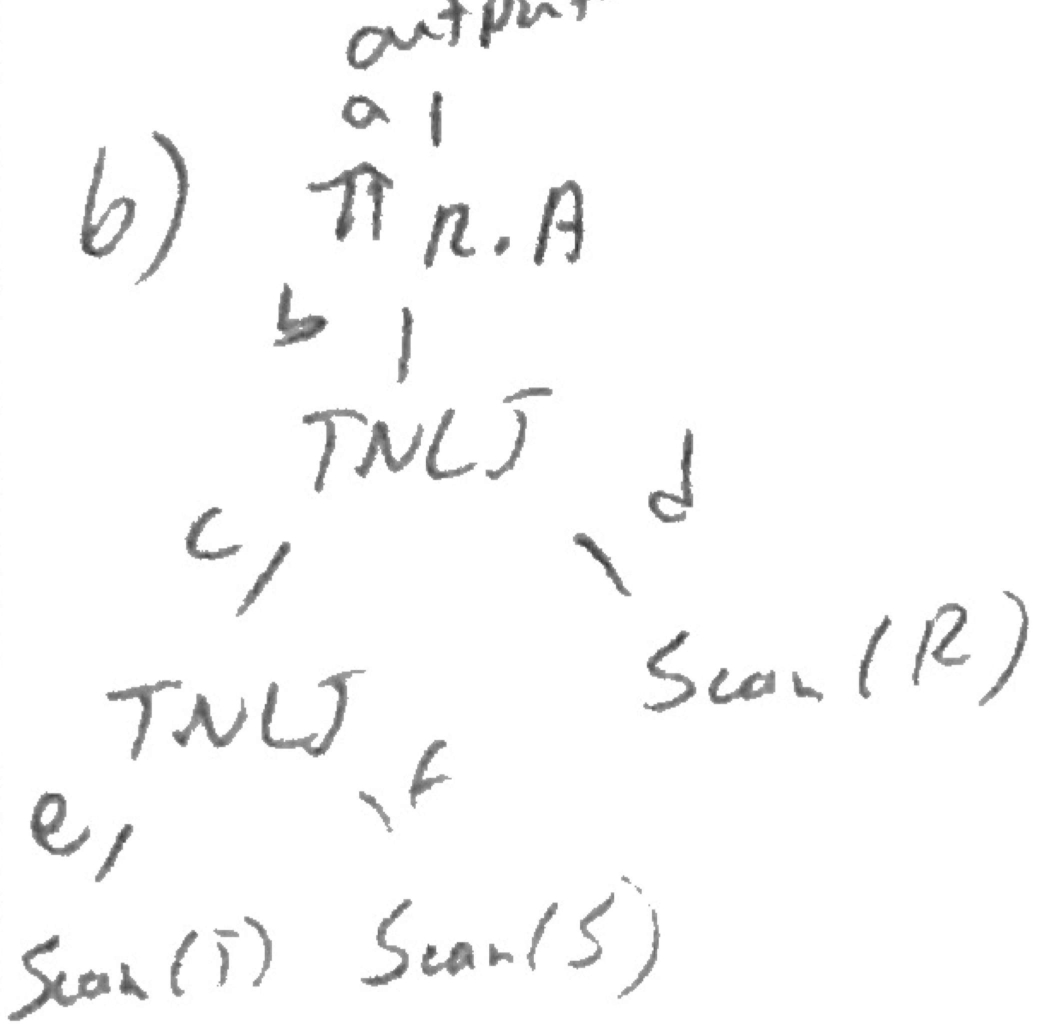
Edge d: $4(12+1)=52 \rightarrow \{6\times(1,1,1),6\times(2,2,2)\}$

Edge b: $12+1=13 \rightarrow \{1,2\}$

Edge a: $2+1=3$

$\boxed{\text{98 GetNext calls}}$

b)




Edge e: $12+1=13$

Edge f: $12(2+1)=36 \rightarrow \{3\times(1,1),3\times(2,2)\}$

Edge c: $6+1=7$

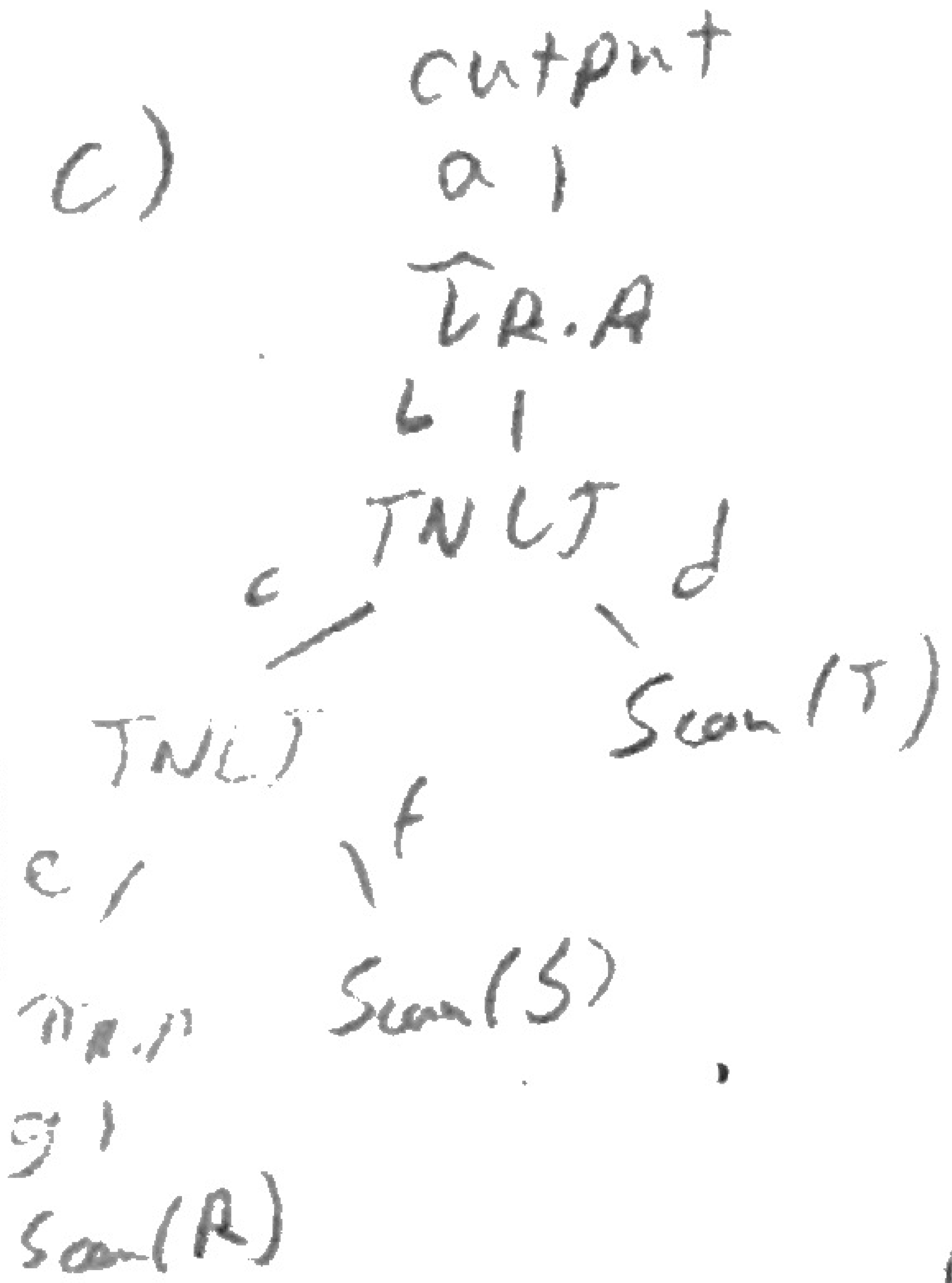
Edge d: $6(6+1)=42 \rightarrow \{6\times(1,1,1),6\times(2,2,2)\}$

Edge b: $12+1=13 \rightarrow \{1,2\}$

Edge a: $2+1=3$

$\boxed{\text{114 GetNext calls}}$

c)




Edge g: $6+1=7 \rightarrow \{1,2,3\}$

Edge e: $3+1=4$

Edge f: $3(2+1)=9 \rightarrow \{(1,1),(2,2)\}$

Edge c: $2+1=3$

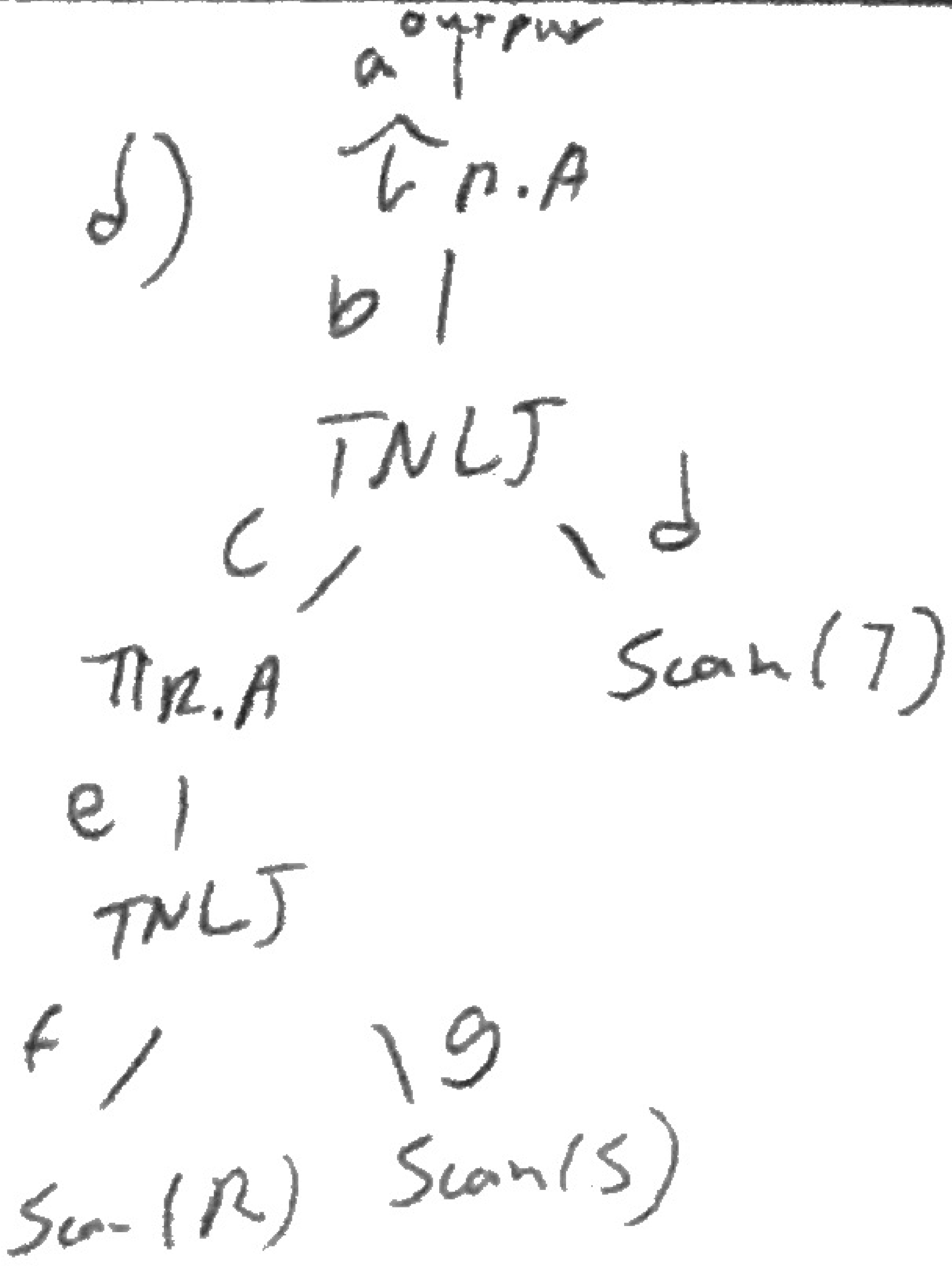
Edge d: $2(12+1)=26 \rightarrow \{3\times(1,1,1),3\times(2,2,2)\}$

Edge b: $6+1=7 \rightarrow \{3\times1,3\times2\}$

Edge a: $6+1=7$

$\boxed{\text{63 GetNext calls}}$

d)




Edge f: $6+1=7$

Edge g: $6(2+1)=18 \rightarrow \{(1,1),(1,1),(2,2),(2,2)\}$

Edge e: $4+1=5 \rightarrow \{1,2\}$

Edge c: $2+1=3$

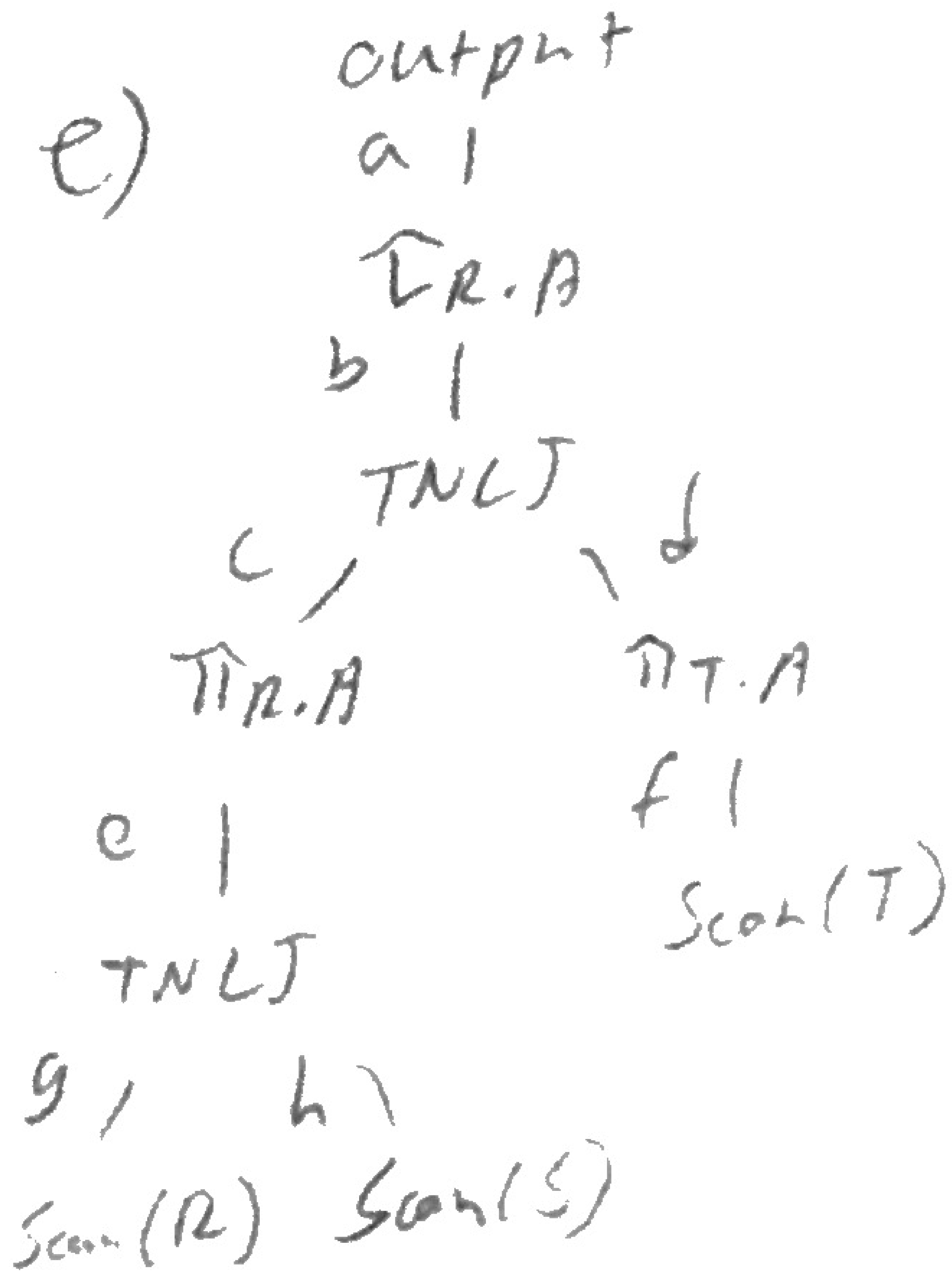
Edge d: $2(12+1)=26 \rightarrow \{3\times(1,1,1), 3\times(2,2,2)\}$

Edge b: $6+1=7 \rightarrow \{3\times 1, 3\times 2\}$

Edge a: $6+1=7$

$\boxed{\text{73 GetNext calls}}$

e)




Edge g: $6+1=7$

Edge h: $6(2+1)=18 \rightarrow \{2\times(1,1), 2\times(2,2)\}$

Edge e: $4+1=5 \rightarrow \{1,2\}$

Edge f: $12+1=13 \rightarrow \{1,2,3,4\}$

Edge c: $2+1=3$

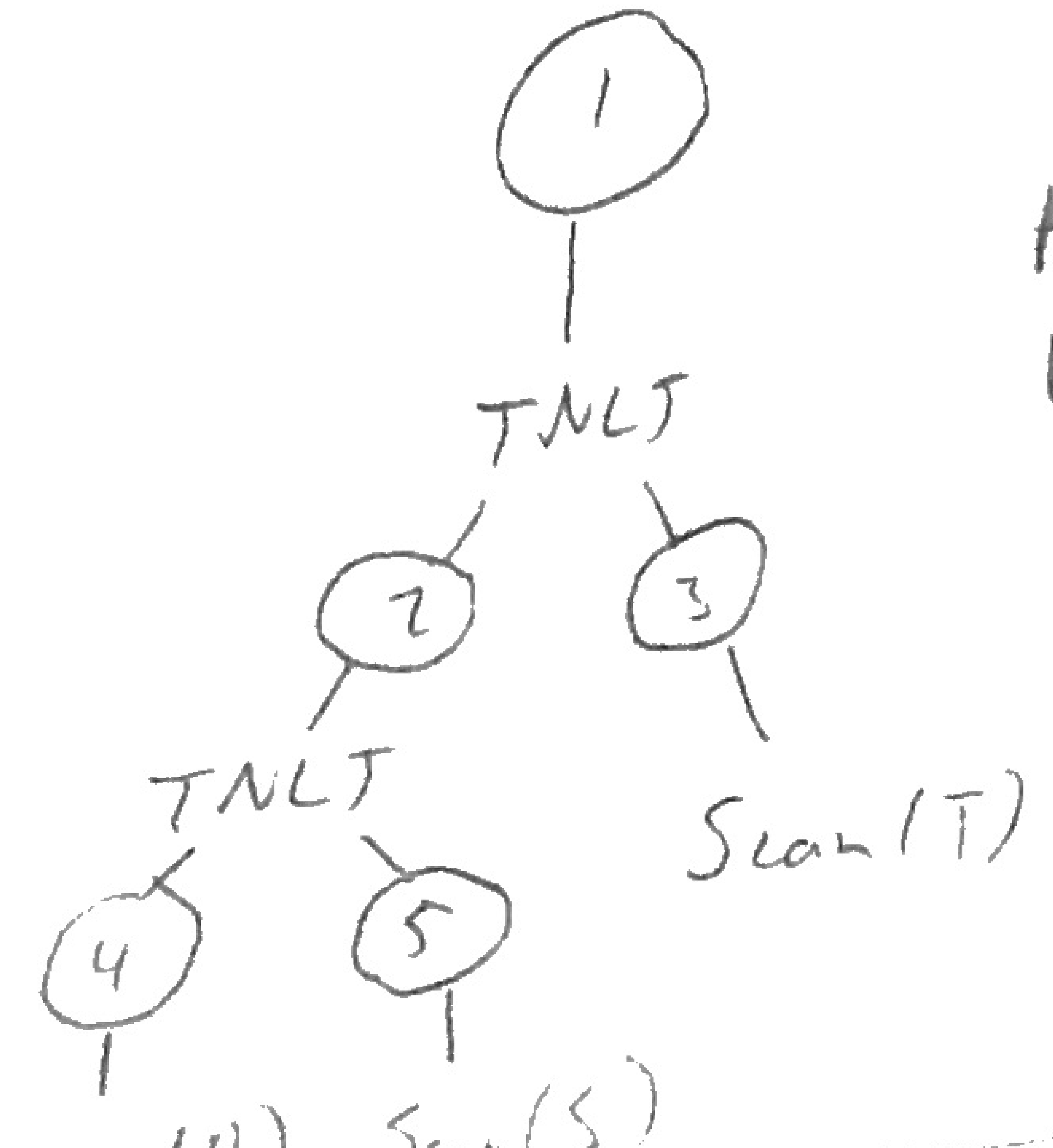
Edge d: $2(4+1)=10 \rightarrow \{(1,1),(2,2)\}$

Edge b: $2+1=3 \rightarrow \{1,2\}$

Edge a: $2+1=3$

$\boxed{\text{62 GetNext calls}}$

3) For a left-deep tree, there are 6 possible permutations of Scan(R), Scan(S), and Scan(T)'s placement. Considering right-deep trees will multiply the number of possible plans by 2. Each projection operator can go in 5 spots or not be there at all ★ top level must have a projection, can't be empty. This gives an upper bound of $6 \times 2 \times 3^4 \times 2 = 1{,}944$ physical plans. However, many of these plans are invalid considering the SQL query in Question 1. An example of an invalid plan included in the 1,944 plans calculated above, is a plan with no duplicate-eliminating projections. Possible spots for projections are shown below.



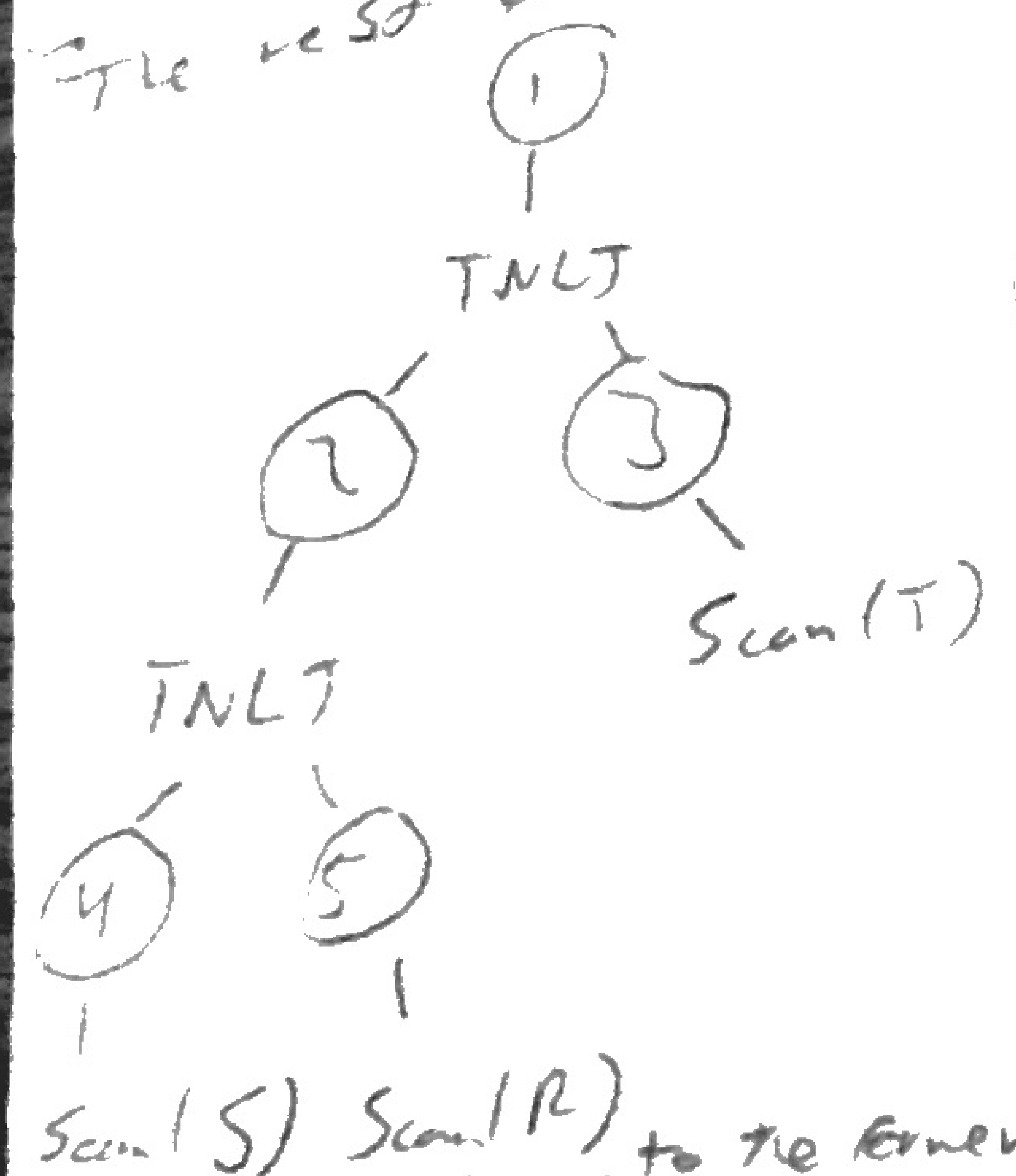

Considering the scans in this position, every physical plan must have a duplicate-eliminating projection in either spots 2 AND 3, 4 AND 3, or 1. Within each of these plans, the rest of the spots can be either a duplicate-preserving or eliminating projection, or it might be blank.

1 case: $3\times3\times3\times3 = 81$

2 AND 3 case: $3\times3 = 9$ → not including duplicates from 1 case

4 AND 3 case: $3\times2 = 6$ → not including dups from 1 case and 2 AND 3 case

The rest of the permutations for Scan(R), Scan(S), and Scan(T) are similarly calculated below.

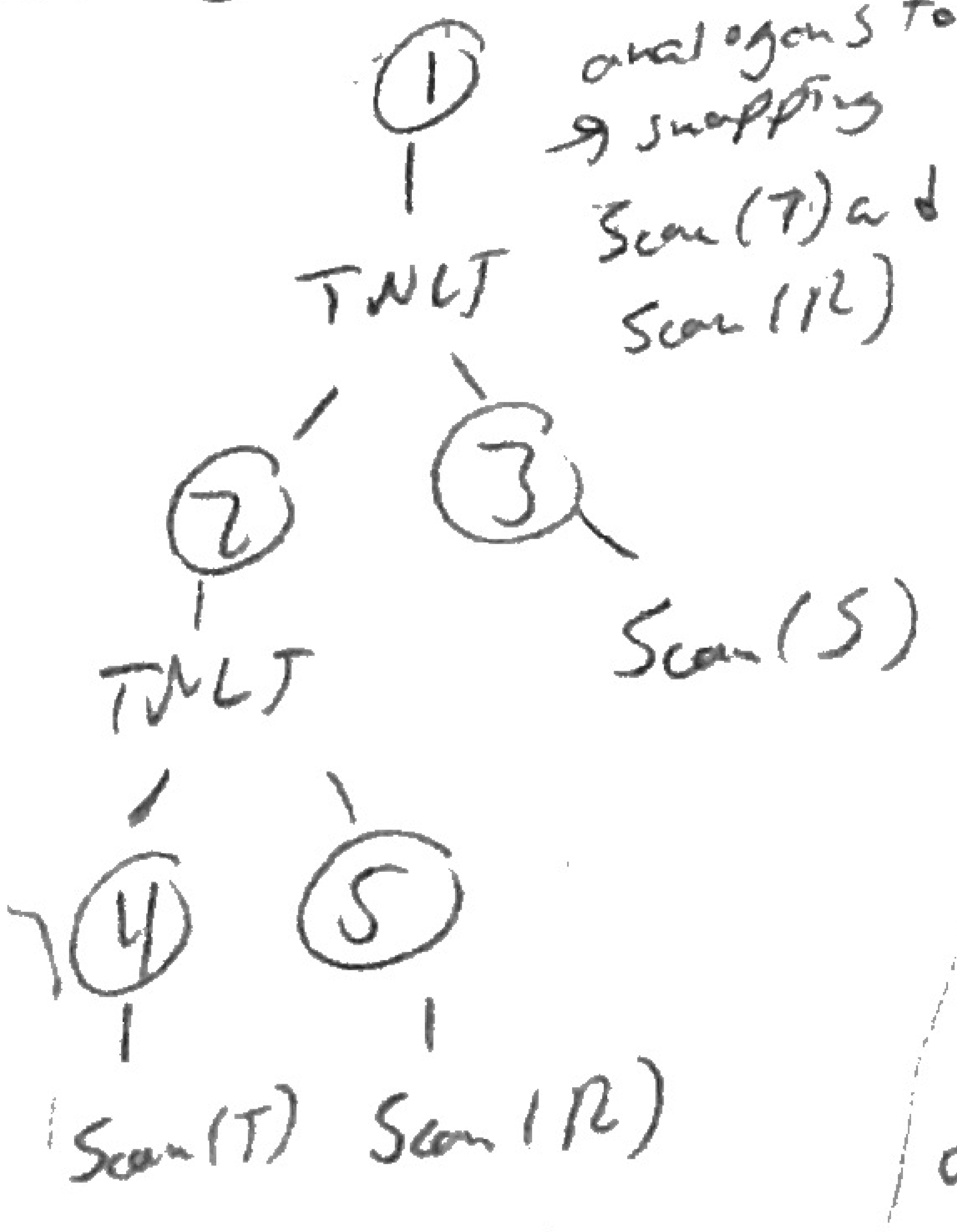



to the former: This is analogous

1 case: 81 permutations

2 AND 3 case: +9

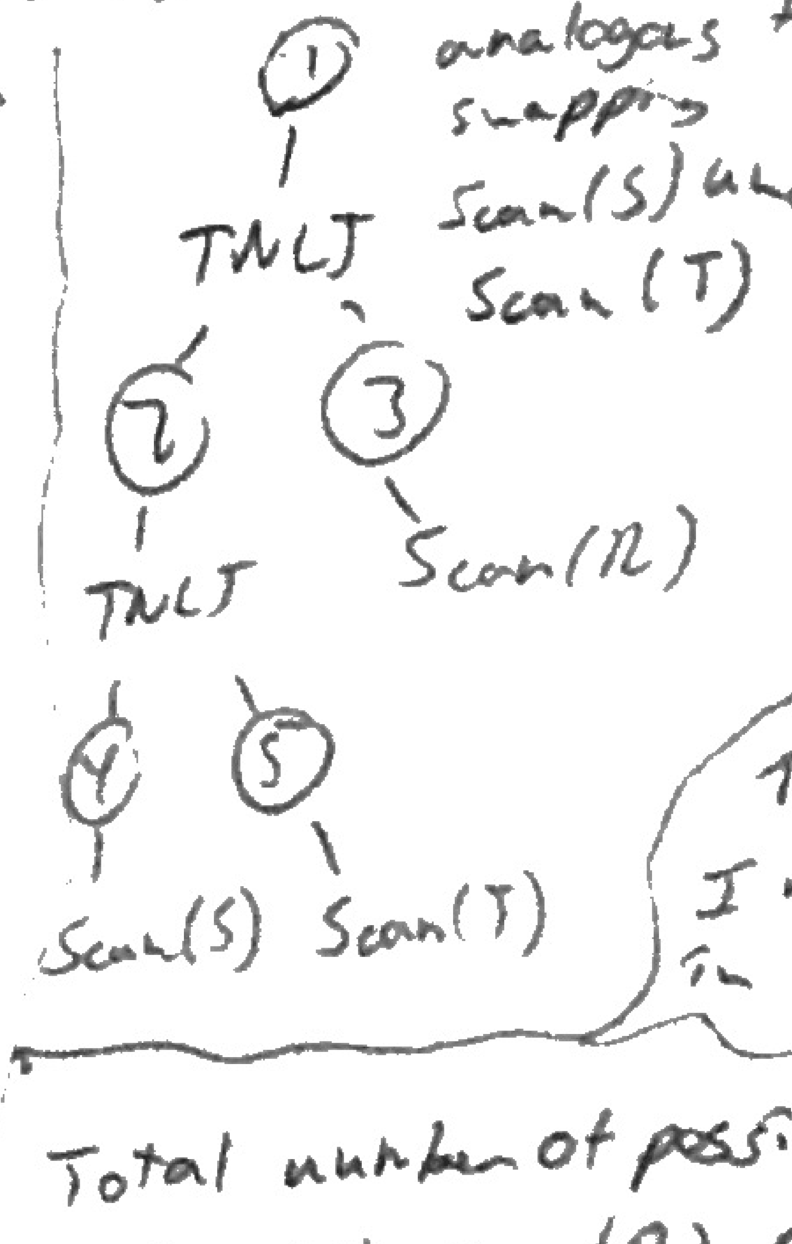
5 AND 3 case: +6




1 case: 81

2 case: +27

4 AND 5 case: +6




1 case: 81

2 AND 3 case: +9

5 AND 3 case: +6

★ Purposely didn't consider projection operators like $\pi_{R.A, S.A}$. This is an assumption I made because it wasn't clear in the assignment writeup

Total number of possible plans for 6 permutations of Scan(S), Scan(R), Scan(T) =

$2(81+9+6)+2(81+27+6)+2(81+9+6)$

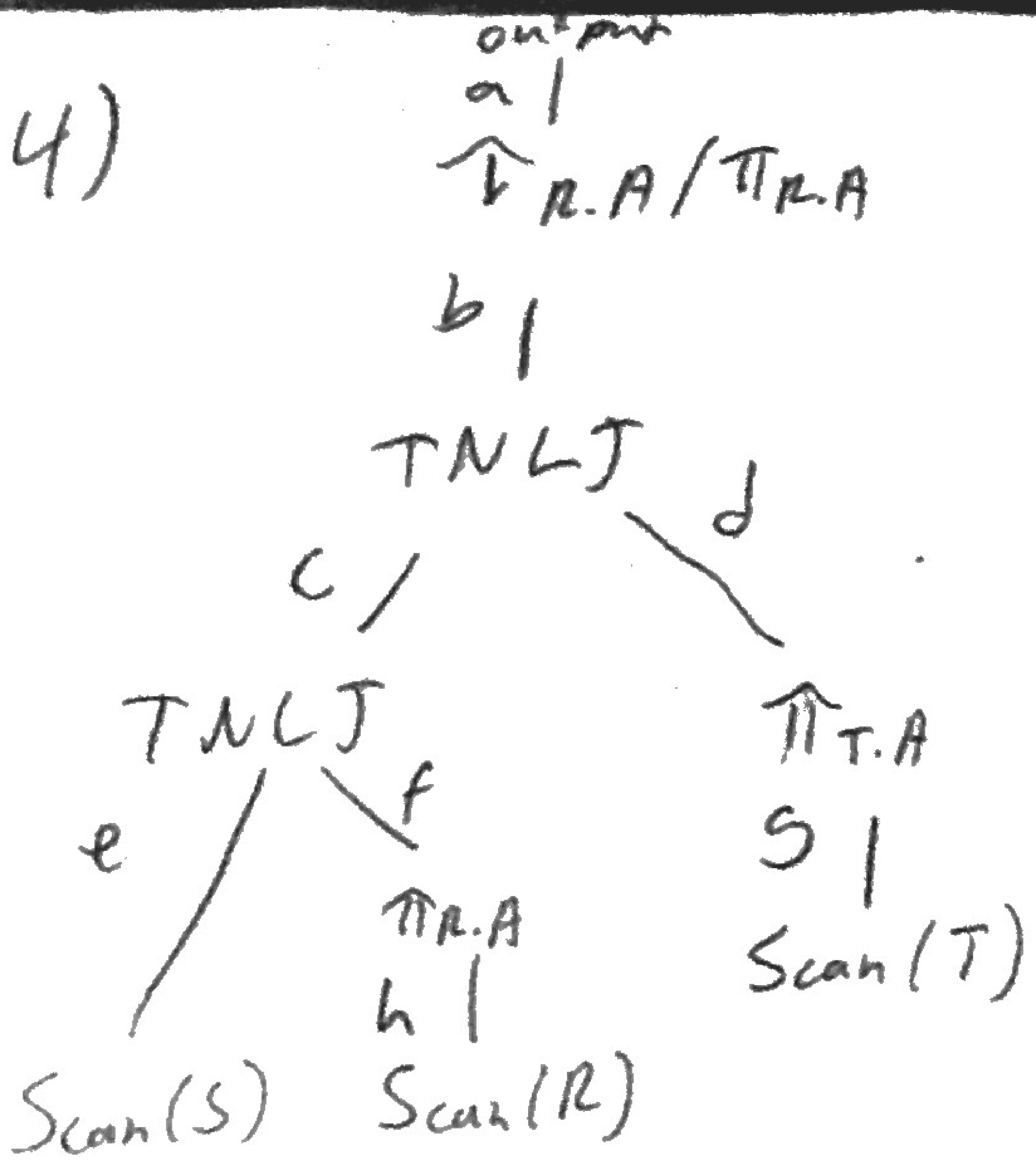
$= 612$

Taking into account right-deep subtrees

$612 \times 2 = \boxed{1{,}224 \text{ physical plans}}$

bounded by 1,944 corroborating this answer

4)




- duplicate-eliminating projects on R and T early to save GetNext calls later in the TNLJs
- smaller tables on left of TNLJ because the TNLJ algorithm leads to $(|L|)(|R|+1)+1$ GetNext calls when L is left table and R is right table. Want to minimize the L tables size for $(|R|)(|L|+1)+1 - (|L|)(|R|+1)+1 = |R|-|L|$ less GetNext calls, when R is the bigger table.
- left-deep subtree over right-deep subtree

Edge e: $2+1=3$

Edge h: $6+1=7 \rightarrow \{1,2,3\}$

Edge f: $2(6+1)=14 \rightarrow \{(1,1),(2,2)\}$

Edge g: $12+1=13 \rightarrow \{1,2,3,4\}$

Edge c: $2+1=3$

Edge d: $2(4+1)=10 \rightarrow \{(1,1,1),(2,2,2)\}$

Edge b: $2+1=3 \rightarrow \{1,2\}$

Edge a: $2+1=3$

$\boxed{\text{56 GetNext calls}}$

Having a lower number of GetNext calls than all the valid plans in 2 corroborates this answer.

5) a) If $(\sigma_{p_1} R_1) \bowtie (\sigma_{p_2} R_2) == \emptyset$ then the equivalence holds.

This makes sense intuitively because if applying the predicates to their respective tables yields two tables that can be joined to produce a table that is anything other than the null set, i.e. the two predicates yield two tables with at least 1 tuple from each having the same B value, then the resulting tuples will be double counted by the terms $(\sigma_{p_1} R_1 \bowtie R_2)$ and $(R_1 \bowtie \sigma_{p_2} R_2)$.

b) It doesn't. The argument above still holds if $R_1$ and $R_2$ can have duplicate tuples in them, so this answer makes sense.

c) $\sigma_{p_1 \vee p_2}(R_1 \bowtie R_2) = (\sigma_{p_1} R_1 \bowtie R_2) \cup_B (R_1 \bowtie \sigma_{p_2} R_2) - (\sigma_{p_1} R_1 \bowtie \sigma_{p_2} R_2)$

$= \sigma_{p_1}(R_1 \bowtie R_2) \cup_B \sigma_{p_2}(R_1 \bowtie R_2) - \sigma_{p_1 \wedge p_2}(R_1 \bowtie R_2)$

This makes sense intuitively because it performs a bag-union for the tuples that satisfy predicate 1 of the natural join of $R_1$ and $R_2$ and the tuples that satisfy predicate 2 of the natural join of $R_1$ and $R_2$. This will have duplicates that the left side won't for the tuples of $R_1 \bowtie R_2$ that satisfy both predicates, which is why a set-difference is performed with $(\sigma_{p_1} R_1 \bowtie \sigma_{p_2} R_2)$.

2) a) $Scan(R) = |R| + 1 = 7$ GetNext calls

$Scan(S) = |S| + 1 = 3$ GetNext calls

$TNLJ_{R,S} = |R|(|S|+1) + 1 = 19$ GetNext calls $\rightarrow \{(1,1),(1,1),(2,2),(2,2)\}$

$Scan(T) = |T| + 1 = 13$ GetNext calls

$TNLJ_{R \bowtie S, T} = |R \bowtie S|(|T|+1) + 1 = 53$ GetNext calls $\rightarrow \{(1,1,1),(1,1,1),(1,1,1),(1,1,1),(1,1,1),(1,1,1),$
$(2,2,2),(2,2,2),(2,2,2),(2,2,2),(2,2,2),(2,2,2)\}$

$\pi_{R.A} = |(R \bowtie S) \bowtie T| + 1 = 13$ GetNext calls $\rightarrow \{1,2\}$ $\boxed{111 \text{ GetNext calls}}$

output $= 2 + 1 = 3$ GetNext calls

b) $Scan(T) = 13$

$Scan(S) = 3$

$TNLJ_{T,S} = (12)(3) + 1 = 37 \rightarrow \{(1,1),(1,1),(1,1),(2,2),(2,2),(2,2)\}$

$Scan(R) = 7$

$TNLJ_{T \bowtie S, R} = (6)(7) + 1 = 43 \rightarrow \{6 \times (1,1,1), 6 \times (2,2,2)\}$

$\pi_{R.A} = 13 \rightarrow \{1,2\}$ $\boxed{119 \text{ GetNext calls}}$

output $= 3$

$\boxed{\text{OLD WAY}}$

c) $Scan(R) = 7$

$\pi_{R.A} = 7 \rightarrow \{1,2,3\}$

$Scan(S) = 3$

$TNLJ_{R,S} = (3)(3) + 1 = 10 \rightarrow \{(1,1),(2,2)\}$

$Scan(T) = 13$

$TNLJ_{R \bowtie S, T} = (2)(13) + 1 = 27 \rightarrow \{3 \times (1,1,1), 3 \times (2,2,2)\}$

$\pi_{R.A} = 7 \rightarrow \{1,1,1,2,2,2\}$ $\boxed{81 \text{ GetNext calls}}$

output $= 7$

d) Scan(R) = 7

Scan(S) = 3

TNLJ = 19 $\rightarrow$ {(1,1),(1,1),(2,2),(2,2)}

$\pi_{R.A}$ = 5 $\rightarrow$ {1,2}

Scan(T) = 13

TNLJ = (2)(13)+1 = 27 $\rightarrow$ {3×(1,1),3×(2,2)}

$\tau_{R.A}$ = 7 $\rightarrow$ {1,1,1,2,2,2}

$\boxed{\text{88 GetNext calls}}$

output = 7

e) Scan(R) = 7

Scan(S) = 3

TNLJ = 19 $\rightarrow$ {(1,1),(1,1),(2,2),(2,2)}

$\pi_{R.A}$ = 5 $\rightarrow$ {1,2}

Scan(T) = 13

$\pi_{T.A}$ = 13 $\rightarrow$ {1,2,3,4}

TNLJ = (2)(5)+1 = 11 $\rightarrow$ {(1,1),(2,2)}

$\tau_{R.A}$ = 3 $\rightarrow$ {1,2}

$\boxed{\text{77 GetNext calls}}$

output = 3 $\rightarrow$ {1,2}

$\boxed{\text{OLD WAY}}$