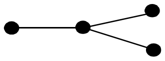
CS386D Problem Set #1

[1] Consider the following query:

```
select * from A a, B b, C c, D d
where a.X=b.X and b.Y=c.Y and c.Z=d.Z and d.W=A.W
```

(a) As shown in class (with the  query graph), list the set of the logical access plans are examined by the System R optimizer. Hint: *do not show the stream ordering and join predicate parameters in your expressions. Follow the analysis in the class notes (choose a sink and find all 1-relation queries, then prune, 2-relation queries, then prune, etc.)*

(b) What logical access plans are *not* examined by the System R optimizer? Why are they are not considered?

[2] Consider a linear query graph. What is the size of the search space that System R examines? (or how many plans does System R generate)? Pick one question — they have different answers.

[3] Consider the following attributes, their cardinalities, and index storage structures:

Attribute	Cardinality	Storage Structure
A	20	B+ trees
B	2000	B+ trees
C	2000	hash
D	20	Not Indexed

Now consider the following local predicates. For each predicate, what index would you use (if any) to most efficiently retrieve the tuples that satisfy this predicate:

(a) B=3 or B=4

(b) B= 66 and C=12

(c) B>3 and C>77

(d) B=22 and A = 15

(e) D=44 and B>34

[4] Suppose join predicates are of the form “A or B or C or ...” where A, B, C, ... are typical conjunctive join predicates. How would you generalize the System R algorithm to process such queries?