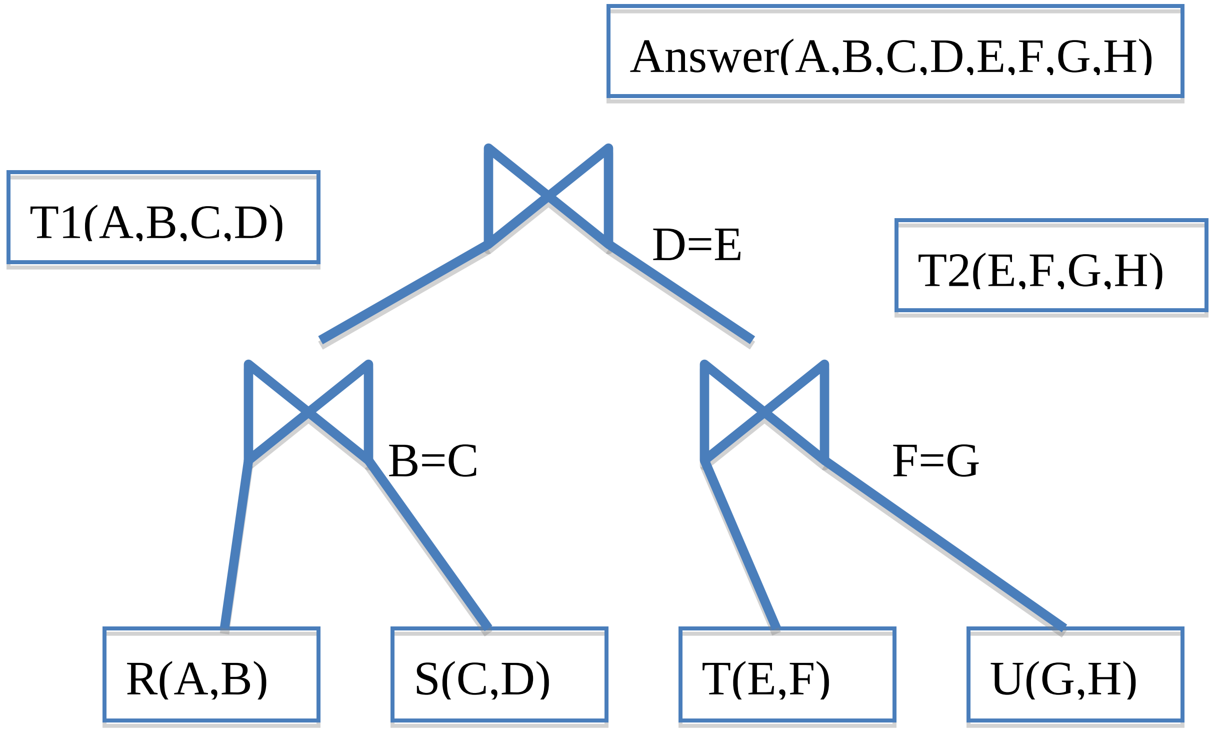
# Section 9

1. Consider the query R(A,B) join S(C,D) join T(E,F) (the join condition is B=C and D=E). Suppose M =100, and B(R) = 30, B(S) = 200, B(T) = 60, B(R join S) =80, B(S join T) = 50. Design an optimal query plan that uses only main-memory hash join algorithms. Your plan may store intermediate results to disk if necessary.

Load R & T into memory and create hash tables of them. Then read blocks of S one at a time, performing the joins in the following graph. All intermediate results are pipelined.

```
                 |
       HashJoin B=C
    /                    \
   /             HashJoin D=E
  /              /               \
R                T                S
```

2. Consider the algebra plan below. Each of the joint operators is a main memory hash join algorithm, using the Open( ), GetNext( ), Close( ) interface. Assuming that all joins are pipelining, show the execution steps for computing the entire join.



Where R, S, T, U have the following content:

R

| | |
|---|---|
| A1 | B |
| A2 | B |

S

| | |
|---|---|
| B | D1 |
| B | D2 |

T

| | |
|---|---|
| D1 | F |
| D2 | F |

U

| | |
|---|---|
| F | H1 |
| F | H2 |

```
T0.open
        T2.open
                U.open
                U.getNext
                U.getNext
                U.getNext   // got NULL
                U.close
                T.open
        T2.getNext
                T.getNext
        T2.getNext
        T2.getNext
                T.getNext
        T2.getNext
        T2.getNext // got NULL
        T2.close
                T.close
        T1.open
                S.open
                S.getNext
                S.getNext
                S.getNext // got NULL
                S.close
                R.open
T0.getNext
        T1.getNext
                R.getNext
T0.getNext
T0.getNext
        T1.getNext
T0.getNext
T0.getNext
        T1.getNext
                R.getNext
T0.getNext
T0.getNext
        T1.getNext
T0.getNext
T0.getNext
        T1.getNext
                R.getNext // got NULL

T0.close
        T1.close
                R.close
```

(b) **[10 points]** Consider the following query, where ⋈ denotes the natural join:

$$R(A, B) \bowtie S(B, C) \bowtie T(C, D) \bowtie U(D, E)$$

Here we only consider left linear plans

i. How many different left linear plans exist for this query ?

n !

ii. Show two different left linear plans without cartesian products.

(((R join S) join T) join U)
(((T join S) join U) join R)

iii. How many different plans without cartesian product exists for this query ?

2 ^ (n-1)