Solution Report

1 Part 1: Generic Join (GJ)

Logic

The Generic Join (GJ), also known as Worst-Case Optimal Join (WCOJ), evaluates the query by iterating over variables one by one in a specific order (e.g., $A_1, A_2, \ldots, A_6$). At each step, it computes the intersection of valid values for the current variable based on the values of previously bound variables and the relations that contain the current variable.

Pseudocode

```
GenericJoin(Relations, Variables):
  Results = []
  Function Recurse(index, currentTuple):
    if index == |Variables|:
      Results.add(currentTuple)
      return

    var = Variables[index]
    candidates = Intersection of:
      for each R in Relations containing var:
        Project(Select(R where bound vars match currentTuple), var)

    for val in candidates:
      Recurse(index + 1, currentTuple + {var: val})

  Recurse(0, {})
  return Results
```

Asymptotic Running Time

$O(N^{FHW})$, where FHW is the Fractional Hypertree Width of the query. For the triangle query, $FHW = 1.5$. So complexity is $O(N^{1.5})$.

2 Part 2: Generalized Hypertree Width (GHW)

Logic

This algorithm decomposes the query into a hypertree (a tree where nodes are "bags" of relations). For this query, we decompose it into two bags connected by a separator.

- **Bag 1**: $\{R_1, R_2, R_3\}$ covering $\{A_1, A_2, A_3\}$.
- **Bag 2**: $\{R_5, R_6, R_7\}$ covering $\{A_4, A_5, A_6\}$.
- **Separator**: $R_4(A_3, A_4)$.

The algorithm computes the join of each bag using standard pairwise joins, then joins the results.

Pseudocode

```
GHW_Join():
  // Compute Bag 1 using standard join
  Bag1_Result = (R1 join R2) join R3

  // Compute Bag 2 using standard join
  Bag2_Result = (R5 join R6) join R7

  // Join Bags
  Result = Bag1_Result join R4 join Bag2_Result
  return Result
```

Asymptotic Running Time

$O(N^{GHW})$, where GHW is the Generalized Hypertree Width. For the triangle query, $GHW = 2$ (because a triangle cannot be covered by 1 edge, it needs 2). So complexity is $O(N^2)$.

3 Part 3: Fractional Hypertree Width (FHW)

Logic

This algorithm uses the same decomposition as GHW but evaluates each bag using the **Generic Join (WCOJ)** algorithm instead of standard pairwise joins. This avoids the intermediate explosion within the bags.

Pseudocode

```
FHW_Join():
  // Compute Bag 1 using WCOJ
  Bag1_Result = GenericJoin({R1, R2, R3}, {A1, A2, A3})

  // Compute Bag 2 using WCOJ
  Bag2_Result = GenericJoin({R5, R6, R7}, {A4, A5, A6})

  // Join Bags
  Result = Bag1_Result join R4 join Bag2_Result
  return Result
```

Asymptotic Running Time

$O(N^{FHW})$. Since the FHW of the triangle sub-query is 1.5, the complexity is dominated by computing the bags: $O(N^{1.5})$.

4 Part 4: Experimental Comparison

I ran the algorithms on a synthetic dataset with $N = 2000$ tuples per relation. The code can be found in `Solution.js`.

Algorithm	Running Time	Asymptotic Complexity
Generic Join (WCOJ)	~ 2.92s	$O(N^{1.5})$
GHW Join	~ 1.43s	$O(N^2)$
FHW Join	~ 2.08s	$O(N^{1.5})$

Analysis

- **Theory vs Practice**: Theoretically, GHW ($O(N^2)$) should be slower than GJ/FHW ($O(N^{1.5})$).
- **Observation**: In this specific experiment, GHW was actually the fastest.
- **Explanation**:
  - **Constant Factors**: The WCOJ implementation in JavaScript involves significant overhead (iterating domains, intersecting sets, recursive calls) compared to the highly optimized hash lookups of a standard pairwise join.
  - **Dataset Properties**: The synthetic dataset, while dense, did not trigger a catastrophic "triangle explosion" where the intermediate size of $(R_1 \bowtie R_2)$ was significantly larger than the final triangle count. If $(R_1 \bowtie R_2)$ is similar in size to $(R_1 \bowtie R_2 \bowtie R_3)$, then the standard join is very efficient. To see the asymptotic benefit of WCOJ, one typically needs a dataset where $|R_1 \bowtie R_2| \gg |R_1 \bowtie R_2 \bowtie R_3|$.
  - **FHW vs GJ**: FHW was faster than pure GJ, likely because decomposing the problem into smaller bags reduced the depth of recursion and allowed for efficient caching of bag results.