**Migration Analysis of Library Versions:**

I performed an analysis to understand how artifacts (projects or systems) are migrating between different versions of the **org.jgrapht:jgrapht-core** library, using data from the Neo4j Maven Central dependency graph.

1. **Identified Migrations**:
   * I tracked the migration patterns from older to newer versions of the library. For instance, I checked how many artifacts migrated from version **0.9.0** to versions like **0.9.1**, **0.9.2**, and **1.0.0**.
2. **Data Representation**:
   * I exported the migration data, showing the **old version**, **new version**, and the number of migrations (i.e., how many artifacts updated from one version to the next).
   * For example, the data shows that **1 artifact migrated from version 0.9.0 to 0.9.1**, and similarly, 1 artifact migrated from **0.9.0 to 1.0.0**.
3. **Visualization**:
   * I used Excel to create a **bar chart** that visualizes the migration counts between different versions of the library.
   * In the chart, the **X-axis** represents the migration paths (e.g., 0.9.0 -> 0.9.1), and the **Y-axis** represents the **number of migrations** (which is 1 for each migration in this case).

**Key Observations:**

1. **Each version migration has exactly 1 artifact** migrating to the next version.
   * For instance, we saw 1 artifact migrate from **0.9.0 to 0.9.1**, and similarly from **0.9.0 to 0.9.2** and **1.0.0**.
2. **There is a clear progression across versions**, with migrations happening from the very early versions (e.g., 0.9.0) to more recent versions like 1.5.2.
3. **No major jumps in the migration count** between versions, meaning that the migration happens at a steady rate across different versions, with artifacts updating gradually from one version to the next.