

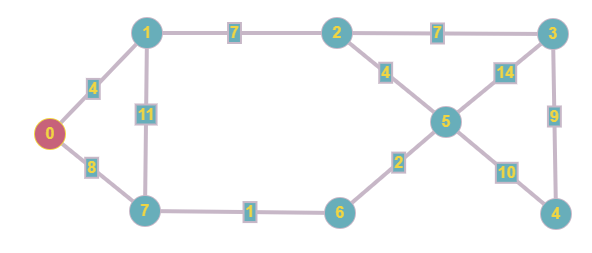
**TCP2101 – Algorithm Design And Analysis**

Group Members (TC03 | TT08):

1. Samuel Wong (1151103687)
2. Wayile Jialade (1151102347)

**Prim MST Demo1:-**

Before:-



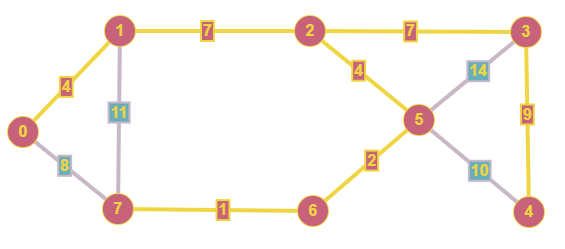
**Adjacency Matrix:-**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Node\Node | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **0** | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 8 |
| **1** | 4 | 0 | 7 | 0 | 0 | 0 | 0 | 11 |
| **2** | 0 | 7 | 0 | 7 | 0 | 4 | 0 | 0 |
| **3** | 0 | 0 | 7 | 0 | 9 | 14 | 0 | 0 |
| **4** | 0 | 0 | 0 | 9 | 0 | 10 | 0 | 0 |
| **5** | 0 | 0 | 4 | 14 | 10 | 0 | 2 | 0 |
| **6** | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 |
| **7** | 8 | 11 | 0 | 0 | 0 | 0 | 1 | 0 |

**Adjacency List:-**

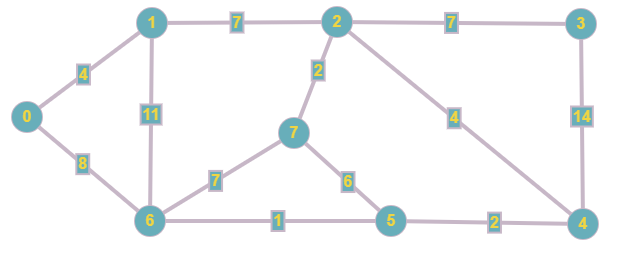
|  |  |  |
| --- | --- | --- |
| **Source** | **Destination** | **Weight** |
| 0 | 1 | 4 |
| 0 | 7 | 8 |
| 1 | 2 | 7 |
| 1 | 7 | 11 |
| 2 | 3 | 7 |
| 2 | 5 | 4 |
| 3 | 4 | 9 |
| 3 | 5 | 14 |
| 4 | 5 | 10 |
| 5 | 6 | 2 |
| 6 | 7 | 1 |

After:-



**Prim MST Demo2:-**

Before:-



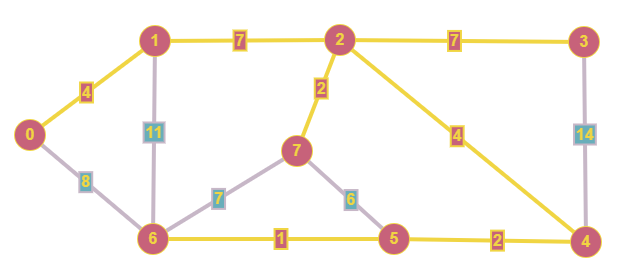
**Adjacency Matrix:-**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Node\Node | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | | **0** | 0 | 4 | 0 | 0 | 0 | 0 | 8 | 0 | | **1** | 4 | 0 | 7 | 0 | 0 | 0 | 11 | 0 | | **2** | 0 | 7 | 0 | 7 | 4 | 0 | 0 | 2 | | **3** | 0 | 0 | 7 | 0 | 14 | 0 | 0 | 0 | | **4** | 0 | 0 | 4 | 14 | 0 | 2 | 0 | 0 | | **5** | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 6 | | **6** | 8 | 11 | 0 | 0 | 0 | 1 | 0 | 7 | | **7** | 0 | 0 | 2 | 0 | 0 | 6 | 7 | 0 | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

**Adjacency List:-**

|  |  |  |
| --- | --- | --- |
| **Source** | **Destination** | **Weight** |
| 0 | 1 | 4 |
| 0 | 6 | 8 |
| 1 | 2 | 7 |
| 1 | 6 | 11 |
| 2 | 3 | 7 |
| 2 | 4 | 4 |
| 2 | 7 | 2 |
| 3 | 4 | 14 |
| 4 | 5 | 2 |
| 5 | 6 | 1 |
| 5 | 7 | 6 |
| 6 | 7 | 7 |

After:-



**What is the difference between dense and sparse graph?**

Dense graph is a graph in which the number of edges is close to the maximal number of edges. Sparse graph is a graph in which the number of edges is close to the minimal number of edges.

**Strategy of testing (Dense):-**

We will try to make number of edges as much as possible

Number of edges = Number of nodes2 - Number of nodes

**Strategy of testing (Sparse):-**

If a number 0 is generated between node A and node B, which means that there is no edge between A and B, therefore for sparse testing, we will always generate number 0 between each nodes, so that there could be fewer edges.

*Note:- use java –Xmx2g filename while out of memory*

**Adjacency Matrix vs Adjacency List (Dense):-**

|  |  |  |
| --- | --- | --- |
| **Number of nodes** | **Adjacency Matrix(seconds)** | **Adjacency List(seconds)** |
| 10 | <1s | <1s |
| 100 | <1s | <1s |
| 1000 | <1s | <1s |
| 2000 | <1s | 2s |
| 3000 | <1s | 5s |
| 4000 | <1s | 12s |
| 5000 | <1s | 26s |
| 6000 | <1s | 40s |
| 10000 | 3s | Out of Memory |

**Adjacency Matrix vs Adjacency List (Sparse):-**

|  |  |  |
| --- | --- | --- |
| **Number of nodes** | **Adjacency Matrix(seconds)** | **Adjacency List(seconds)** |
| 10 | <1s | <1s |
| 100 | <1s | <1s |
| 1000 | <1s | <1s |
| 5000 | <1s | <1s |
| 10000 | 3s | <1s |
| 20000 | 23s | 1s |
| 30000 | 234s | 1s |
| 50000 | Out of Memory | 3s |
| 100000 | Out of Memory | 9s |

**Conclusion:-**

In Dense Graph, The Adjacency Matrix is preferred.

In Sparse Graph, The Adjacency List is preferred.