To run the code, do the following:

>> javac Display.java

>> java Display

Comparison of the 2 algorithms for TSP:

Below is a table with tests for many values of n.

We observe that the Nearest Neighbor algorithm always completes very fast, almost instantly. This is because it computes in O(n2) time. On the other hand, brute force takes increasingly more time for larger values of n. This is because it computes in O(n!) time.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Testing both Algorithms (NOTE: time in ms) | | | | | | | | |
| n | Test 1 | | Test 2 | | Test 3 | | Average | |
| Nearest Neighbor | Brute Force | Nearest Neighbor | Brute Force | Nearest Neighbor | Brute Force | Nearest Neighbor | Brute Force |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00 | 0.00 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00 | 0.00 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00 | 0.00 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00 | 0.00 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00 | 0.00 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00 | 0.00 |
| 8 | 0 | 7 | 0 | 8 | 0 | 7 | 0.00 | 7.33 |
| 9 | 0 | 45 | 0 | 45 | 0 | 46 | 0.00 | 45.33 |
| 10 | 0 | 656 | 0 | 652 | 0 | 675 | 0.00 | 661.00 |
| 11 | 0 | 5161 | 0 | 5239 | 0 | 5487 | 0.00 | 5295.67 |