In this exercise I was to solve the Bandwidth Coloring Problem. I chose to do this three different ways where they all produce feasible solutions.

## Method 1 - Adapted Prim's Algorithm with Multistart

The first method I used is loosely based on Prim's algorithm with a multistart approach. I started by choosing a starting node randomly, and expanded with the edge that had the lowest available cost. This method has a runtime of $\mathcal{O}(nml)$, where $n$ is the number of nodes, $m$ is the number of edges, and $l$ is the number iterations. Therefore, this is a quite slow method, but you are guaranteed an *ok* solution.

## Method 2 - Adapted Kruskal's Algorithm

This method was based on Kruskal's Algorithm, i.e. I evaluated the edge with the lowest cost of all the edges if one of the connected nodes hadn't been visited yet. This had a total runtime of $\mathcal{O}(nm)$, where $n$ was the number of nodes, $m$ was the number of edges. This is a rather quick method, but it comes on the cost of a guaranteed optimal solution.

## Method 3 - Method of Randomized Edges with Multistart

The last method I implemented was an algorithm that chose a random edge where one of the nodes hadn't been visited yet, and calculated the lowest possible coloring for the nodes. To ensure good results, I used the multistart method. The runtime was therefore $\mathcal{O}(nml)$, where $n$ was the number of nodes, $m$ was the number of edges, and $l$ was the number iterations. However, it still runs quicker than Method 1 and gives a better solution given that you have enough iterations in the multistart.

## Results

One can see that Method 3, i.e. the method of randomized edges with multistart, produces the best results overall. Method 2 runs much quicker than the others and still produces quite good results, so it is a good option if speed is an important factor. Method 1 however is quite slow for larger datasets, but produces better results than Method 3 when we have a high amount of nodes. The reason for this is because I've chosen to only look at a few iterations of the multistart for such large datasets, and thus Method 1 has an advantage as it chooses the lowest cost edges, while Method 3 picks edges randomly. Therefore Method 3 won't really be optimized, it will only show a random solution. If one however had run the methods with a higher number of iterations, Method 3 would be the best results wise.

| | Method 1 | Method 2 | Method 3 |
|---|---|---|---|
| GEOM020a | 23 | 25 | 26 |
| GEOM020b | 16 | 16 | 19 |
| GEOM020c | 24 | 26 | 23 |
| GEOM030a | 38 | 41 | 36 |
| GEOM030b | 32 | 42 | 34 |
| GEOM030c | 33 | 35 | 30 |
| GEOM040a | 50 | 60 | 47 |
| GEOM040b | 47 | 58 | 44 |
| GEOM040c | 33 | 36 | 33 |
| GEOM050a | 66 | 73 | 60 |
| GEOM050b | 69 | 83 | 62 |
| GEOM050c | 38 | 44 | 38 |

| | Method 1 | Method 2 | Method 3 |
|---|---|---|---|
| GEOM060a | 71 | 80 | 74 |
| GEOM060b | 81 | 96 | 84 |
| GEOM060c | 40 | 57 | 49 |
| GEOM070a | 85 | 112 | 86 |
| GEOM070b | 96 | 118 | 85 |
| GEOM070c | 56 | 84 | 57 |
| GEOM080a | 115 | 119 | 108 |
| GEOM080b | 109 | 142 | 118 |
| GEOM080c | 62 | 119 | 67 |
| GEOM090a | 120 | 160 | 128 |
| GEOM090b | 144 | 161 | 141 |
| GEOM090c | 74 | 124 | 71 |

| | Method 1 | Method 2 | Method 3 |
|---|---|---|---|
| GEOM0100a | 161 | 206 | 135 |
| GEOM0100b | 157 | 192 | 157 |
| GEOM0100c | 89 | 136 | 87 |
| GEOM0110a | 166 | 220 | 144 |
| GEOM0110b | 185 | 205 | 161 |
| GEOM0110c | 103 | 136 | 86 |
| GEOM0120a | 195 | 243 | 162 |
| GEOM0120b | 212 | 219 | 186 |
| GEOM0120c | 115 | 169 | 92 |
| rand0100a | 147 | 187 | 151 |
| rand0100b | 137 | 152 | 146 |
| rand0100c | 152 | 167 | 163 |
| rand0100d | 144 | 177 | 148 |

| | Method 1 | Method 2 | Method 3 |
|---|---|---|---|
| rand0200a | 346* | 333 | 345 |
| rand0200b | 334* | 391 | 356 |
| rand0200c | 325* | 377 | 356 |
| rand0200d | 309* | 388 | 340 |
| rand0500a | 375* | 511 | 541* |
| rand0500b | 399* | 478 | 471* |
| rand0500c | 390* | 481 | 619* |
| rand0500d | 381* | 498 | 658* |
| rand01000a | - | 837 | 1368* |
| rand01000b | - | 961 | 1293* |
| rand01000c | - | 919 | 1146* |
| rand01000d | - | 945 | 1248* |

Table 1: Results for the Bandwidth Coloring Problem. Did not run Method 1 for the `rand1000`-files as it was incredibly slow.
*Only iterates with 10 times for datasets of this size as it becomes too slow otherwise.