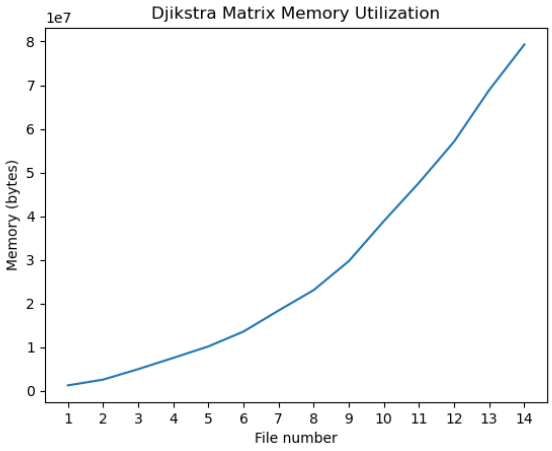
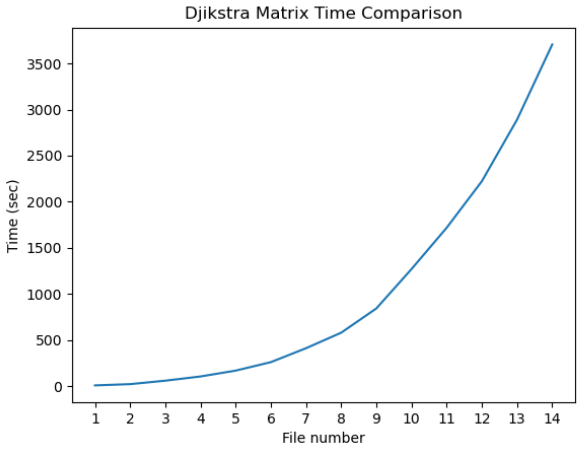
Dijkstra’s Algorithm

1. 2D Array



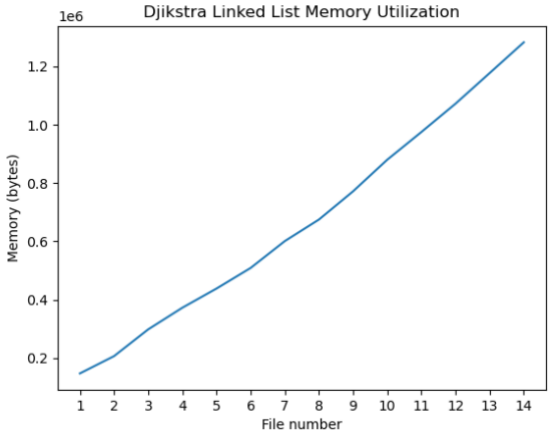


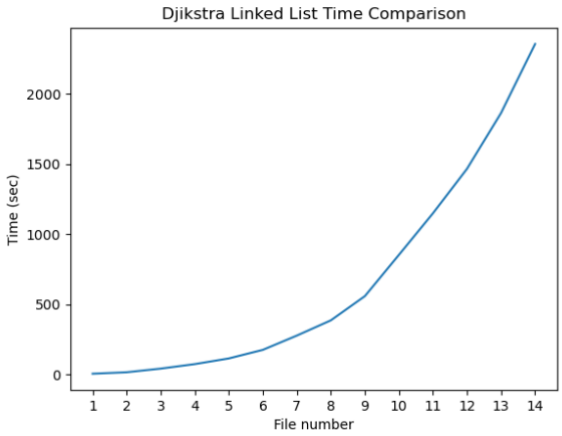
Output:

```
Performance (Seconds)
File 1: 7.616750001907349
File 2: 21.2447350025177
File 3: 58.701220750808716
File 4: 104.3107168674469
File 5: 167.8253688812256
File 6: 259.76771116256714
File 7: 410.7874927520752
File 8: 579.2420518398285
File 9: 842.3586061000824
File 10: 1270.0356318950653
File 11: 1718.109167098999
File 12: 2222.979113340378
File 13: 2890.1213550567627
File 14: 3705.75461101532

Memory utilization (Bytes)
File 1: 1293944
File 2: 2607208
File 3: 5004696
File 4: 7580016
File 5: 10209448
File 6: 13602328
File 7: 18428072
File 8: 23098920
File 9: 29729192
File 10: 38904536
File 11: 47685752
File 12: 57100008
File 13: 68910920
File 14: 79316344
```

```
TEST CASE I
Shortest distance between nodes 192 and 163: 819
Shortest path from nodes 192 to 163:
192 -> 157 -> 194 -> 158 -> 161 -> 163

TEST CASE II
Shortest distance between nodes 138 and 66: 2728
Shortest path from nodes 138 to 66:
138 -> 162 -> 136 -> 159 -> 119 -> 116 -> 114 -> 112 -> 70 -> 110 -> 79 -> 108 -
> 107 -> 103 -> 105 -> 85 -> 67 -> 66

TEST CASE III
Shortest distance between nodes 465 and 22: 6738
Shortest path from nodes 465 to 22:
465 -> 377 -> 380 -> 372 -> 363 -> 364 -> 305 -> 247 -> 210 -> 233 -> 201 -> 170
 -> 128 -> 98 -> 78 -> 41 -> 23 -> 22
```

2. Linked List



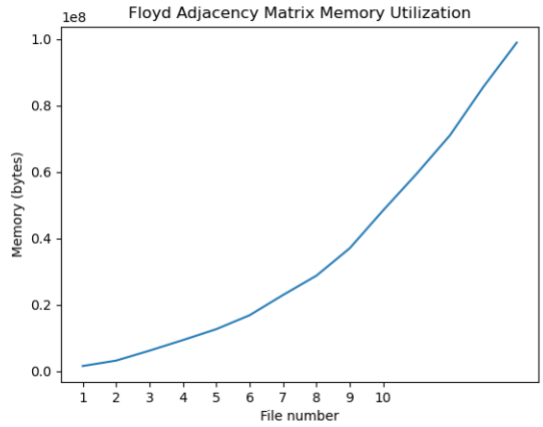


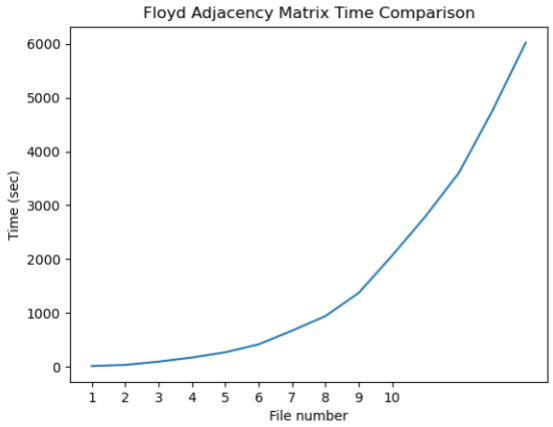
Output:

```
Performance (Seconds)
File 1: 6.921087980270386
File 2: 17.253701210021973
File 3: 43.31109094619751
File 4: 75.33785080909729
File 5: 115.53721308708191
File 6: 176.59407901763916
File 7: 278.5650119781494
File 8: 387.2752151489258
File 9: 559.7402338981628
File 10: 853.7826387882233
File 11: 1150.0094170570374
File 12: 1467.0236449241638
File 13: 1864.793965101242
File 14: 2356.201429843902 6

Memory utilization (Bytes)
File 1: 147206
File 2: 206606
File 3: 298270
File 4: 372510
File 5: 438230
File 6: 508846
File 7: 600886
File 8: 674854
File 9: 771734
File 10: 880022
File 11: 974990
File 12: 1072214
File 13: 1177222
File 14: 1282598
Project2_DijkstraAlgorithm_LinkedList.py:55: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df = df.append(other=pd.read_csv(f), ignore_index=True)

TEST CASE I
Shortest distance between nodes 192 and 163 is: 819
Shortest path traversed from nodes 192 to 163 is:
192 -> 157 -> 194 -> 158 -> 161 -> 163

TEST CASE II
Shortest distance between nodes 138 and 66 is: 2728
Shortest path traversed from nodes 138 to 66 is:
138 -> 162 -> 136 -> 159 -> 119 -> 116 -> 114 -> 112 -> 70 -> 110 -> 79 -> 108 -> 107 -> 103 -> 105 -> 85 -> 67 -> 66

TEST CASE III
Shortest distance between nodes 465 and 22 is: 6738
Shortest path traversed from nodes 465 to 22 is:
465 -> 377 -> 380 -> 372 -> 363 -> 364 -> 305 -> 247 -> 210 -> 233 -> 201 -> 170 -> 128 -> 98 -> 78 -> 41 -> 23 -> 22
(base) rishusingh@Rishus-MacBook-Pro Downloads %
```

Floyd Algorithm

1. 2D Array



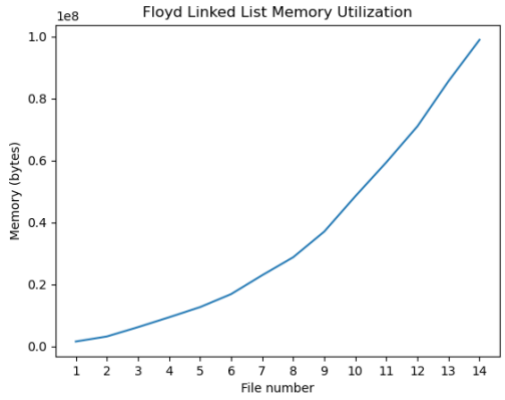


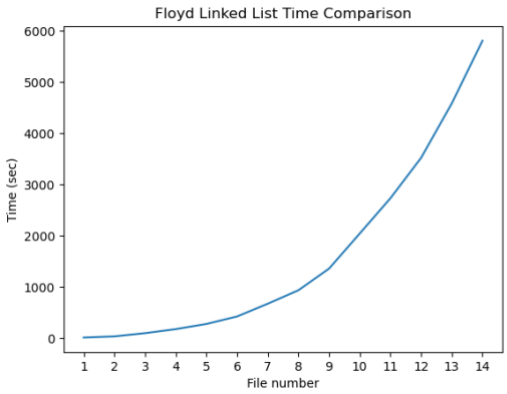
Output:

```
Performance (Seconds)
File 1: 12.17240309715271
File 2: 33.27323007583618
File 3: 93.12424993515015
File 4: 169.4957938194275
File 5: 267.2350170612335
File 6: 415.25745820999146
File 7: 669.1236259937286
File 8: 940.3864860534668
File 9: 1372.3201899528503
File 10: 2066.0251519680023
File 11: 2795.8916580677032
File 12: 3605.844573020935
File 13: 4754.864248991013
File 14: 6024.228240966797

Memory utilization (Bytes)
File 1: 1608088
File 2: 3230680
File 3: 6228440
File 4: 9408448
File 5: 12691928
File 6: 16889880
File 7: 22976552
File 8: 28784392
File 9: 37056088
File 10: 48444568
File 11: 59418104
File 12: 71017928
File 13: 85568584
File 14: 98909528
```

```
TEST CASE I
Shortest distance between nodes 192 and 163: 819
Shortest path from nodes 192 to 163:
192 -> 157 -> 194 -> 158 -> 161 -> 163

TEST CASE II
Shortest distance between nodes 138 and 66: 2728
Shortest path from nodes 138 to 66:
138 -> 162 -> 136 -> 159 -> 119 -> 116 -> 114 -> 112 -> 70 -> 110 -> 79 -> 108 -> 107 -> 103 -> 105 -> 85 -> 67 -> 66

TEST CASE III
Shortest distance between nodes 465 and 22: 6738
Shortest path from nodes 465 to 22:
465 -> 377 -> 380 -> 372 -> 363 -> 364 -> 305 -> 247 -> 210 -> 233 -> 201 -> 170 -> 128 -> 98 -> 78 -> 41 -> 23 -> 22
(base) rishusingh@Rishus-MacBook-Pro Downloads %
```

2. Linked List





Output:

```
Performance (Seconds)
File 1: 13.579736948013306
File 2: 35.8726301932373
File 3: 98.44684076309204
File 4: 176.36490321159363
File 5: 276.39108204841614
File 6: 425.8050639629364
File 7: 672.534903049469
File 8: 937.7432019710541
File 9: 1378.9519050121307
File 10: 2662.554146051407
File 11: 34222.73107481003
File 12: 4815.709204912186
File 13: 5716.071954011917
File 14: 5918.378736257553

Memory utilization (Bytes)
File 1: 1608088
File 2: 3230680
File 3: 6228440
File 4: 9408448
File 5: 12691928
File 6: 16889880
File 7: 22976552
File 8: 28784392
File 9: 37056088
File 10: 48444568
File 11: 59418104
File 12: 71017928
File 13: 85568584
File 14: 98909528
Project2_FloydAlgorithm_LinkedList.py:46: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df = df.append(other=pd.read_csv(f), ignore_index=True)
100%|██████████████████████████████████| 480/480 [02:43<00:00,  2.93it/s]

TEST CASE I
Shortest distance between nodes 192 and 163 is: 819
Shortest path traversed from nodes 192 to 163 is:
192 -> 157 -> 194 -> 158 -> 161 -> 163

TEST CASE II
Shortest distance between nodes 138 and 66 is: 2728
Shortest path traversed from nodes 138 to 66 is:
138 -> 162 -> 136 -> 159 -> 119 -> 116 -> 114 -> 112 -> 70 -> 110 -> 79 -> 108 -> 107 -> 103 -> 105 -> 85 -> 67 -> 66

TEST CASE III
Shortest distance between nodes 465 and 22 is: 6738
Shortest path traversed from nodes 465 to 22 is:
465 -> 377 -> 380 -> 372 -> 363 -> 364 -> 305 -> 247 -> 210 -> 233 -> 201 -> 170 -> 128 -> 98 -> 78 -> 41 -> 23 -> 22
(base) rishusingh@Rishus-MacBook-Pro Downloads %
```

Analysis:

- **Data Structure:** Linked list takes more memory than 2D array as reference to next node is also stored along with the data whereas when it comes to efficiency as per time linked list has a slightly better performance than 2D array.
- **Algorithm:** Dijkstra's algorithm is both time and memory efficient than Floyd algorithm.