

File 1)

```
jehad@jehad-MacBookPro:~/Downloads/Ass3$ make
g++ Dijkstra.cpp Edge.cpp Graph.cpp List_Graph.cpp Matrix_Graph.cpp -o a.out -std=c++11
jehad@jehad-MacBookPro:~/Downloads/Ass3$ ./a.out test1.txt Matrix
```

This is the Standard Dijkstra's Algorithm...  
The Standard Dijkstra's Algorithm for Matrix took 10 microseconds.  
Results for the Standard Dijkstra's Algorithm:

Vertex	Distance from Source	Predecessor
0	0	-1
1	10	0
2	50	3
3	30	0
4	60	2

Running priority queue-based Dijkstra's algorithm...  
Priority Queue-based Dijkstra's algorithm for Matrix took 17 microseconds.  
Results for the Priority Queue-based Dijkstra's Algorithm:

Vertex	Distance from Source	Predecessor
0	0	-1
1	10	0
2	50	3
3	30	0
4	60	2

Comparing results of both implementations:  
Standard Dijkstra's algorithm is faster.

Both implementations produced the same results.

```
jehad@jehad-MacBookPro:~/Downloads/Ass3$ ./a.out test1.txt List
```

This is the Standard Dijkstra's Algorithm...  
The Standard Dijkstra's Algorithm for List took 13 microseconds.  
Results for the Standard Dijkstra's Algorithm:

Vertex	Distance from Source	Predecessor
0	0	-1
1	10	0
2	50	3
3	30	0
4	60	2

Running priority queue-based Dijkstra's algorithm...  
Priority Queue-based Dijkstra's algorithm for List took 21 microseconds.  
Results for the Priority Queue-based Dijkstra's Algorithm:

Vertex	Distance from Source	Predecessor
0	0	-1
1	10	0
2	50	3
3	30	0
4	60	2

Comparing results of both implementations:  
Standard Dijkstra's algorithm is faster.

Both implementations produced the same results.

## File 2)

```
This is the Standard Dijkstra's Algorithm...
The Standard Dijkstra's Algorithm for Matrix took 20 microseconds.
Results for the Standard Dijkstra's Algorithm:
-----
Vertex    Distance from Source  Predecessor
-----
0         0              -1
1         8              3
2         9              1
3         5              0
4         7              0

Running priority queue-based Dijkstra's algorithm...
Priority Queue-based Dijkstra's algorithm for Matrix took 16 microseconds.
Results for the Priority Queue-based Dijkstra's Algorithm:
-----
Vertex    Distance from Source  Predecessor
-----
0         0              -1
1         8              3
2         9              1
3         5              0
4         7              0

Comparing results of both implementations:
Priority Queue-based Dijkstra's algorithm is faster.
-----
Both implementations produced the same results.
```

```
This is the Standard Dijkstra's Algorithm...
The Standard Dijkstra's Algorithm for List took 16 microseconds.
Results for the Standard Dijkstra's Algorithm:
-----
Vertex    Distance from Source  Predecessor
-----
0         0              -1
1         7              3
2         8              1
3         5              0
4         7              0

Running priority queue-based Dijkstra's algorithm...
Priority Queue-based Dijkstra's algorithm for List took 17 microseconds.
Results for the Priority Queue-based Dijkstra's Algorithm:
-----
Vertex    Distance from Source  Predecessor
-----
0         0              -1
1         7              3
2         8              1
3         5              0
4         7              0

Comparing results of both implementations:
Standard Dijkstra's algorithm is faster.
-----
Both implementations produced the same results.
```

## File 3)

```

This is the Standard Dijkstra's Algorithm...
The Standard Dijkstra's Algorithm for Matrix took 20 microseconds.
Results for the Standard Dijkstra's Algorithm:
-----
Vertex    Distance from Source  Predecessor
-----
0         0              -1
1        320             0
2        450             1
3        500             1
4        570             2
5        630             4
6        610             4
7        680             3
8        800             7
9        860             7

Running priority queue-based Dijkstra's algorithm...
Priority Queue-based Dijkstra's algorithm for Matrix took 47 microseconds.
Results for the Priority Queue-based Dijkstra's Algorithm:
-----
Vertex    Distance from Source  Predecessor
-----
0         0              -1
1        320             0
2        450             1
3        500             1
4        570             2
5        630             4
6        610             4
7        680             3
8        800             7
9        860             7

Comparing results of both implementations:
Standard Dijkstra's algorithm is faster.
-----
Both implementations produced the same results.

```

```

This is the Standard Dijkstra's Algorithm...
The Standard Dijkstra's Algorithm for List took 30 microseconds.
Results for the Standard Dijkstra's Algorithm:
-----
Vertex    Distance from Source  Predecessor
-----
0         0              -1
1        320             0
2        450             1
3        500             1
4        570             2
5        630             4
6        610             4
7        680             3
8        800             7
9        860             7

Running priority queue-based Dijkstra's algorithm...
Priority Queue-based Dijkstra's algorithm for List took 17 microseconds.
Results for the Priority Queue-based Dijkstra's Algorithm:
-----
Vertex    Distance from Source  Predecessor
-----
0         0              -1
1        320             0
2        450             1
3        500             1
4        570             2
5        630             4
6        610             4
7        680             3
8        800             7
9        860             7

Comparing results of both implementations:
Priority Queue-based Dijkstra's algorithm is faster.
-----
Both implementations produced the same results.

```

File 4)

```
This is the Standard Dijkstra's Algorithm...
The Standard Dijkstra's Algorithm for Matrix took 44 microseconds.
Results for the Standard Dijkstra's Algorithm:
-----
Vertex    Distance from Source  Predecessor
-----
0          0              -1
1         320              0
2         450              1
3         500              1
4         570              2
5         630              4
6         610              4
7         680              3
8         800              7
9         860              7
10        1000             9
11        1100             10
12        1130             10
13        1250             12
14        1360             13
15        1465             14
16        1510             14
17        1580             15
18        1600             16
19        1660             18
20        1740             18
21        1785             19
22        1885             20
23        1880             21
24        1980             23

Running priority queue-based Dijkstra's algorithm...
Priority Queue-based Dijkstra's algorithm for Matrix took 37 microseconds.
Results for the Priority Queue-based Dijkstra's Algorithm:
-----
Vertex    Distance from Source  Predecessor
-----
0          0              -1
1         320              0
2         450              1
3         500              1
4         570              2
5         630              4
6         610              4
7         680              3
8         800              7
9         860              7
10        1000             9
11        1100             10
12        1130             10
13        1250             12
14        1360             13
15        1465             14
16        1510             14
17        1580             15
18        1600             16
19        1660             18
20        1740             18
21        1785             19
22        1885             20
23        1880             21
24        1980             23

Comparing results of both implementations:
Priority Queue-based Dijkstra's algorithm is faster.
-----
Both implementations produced the same results.
```

```
jehad@jehad-MacBookPro:~/Downloads/Ass3$ ./a.out test4.txt List
```

This is the Standard Dijkstra's Algorithm...

The Standard Dijkstra's Algorithm for List took 172 microseconds.

Results for the Standard Dijkstra's Algorithm:

```
-----  
Vertex    Distance from Source  Predecessor  
-----  
0          0              -1  
1         320              0  
2         450              1  
3         500              1  
4         570              2  
5         630              4  
6         610              4  
7         680              3  
8         800              7  
9         860              7  
10        1000             9  
11        1100             10  
12        1130             10  
13        1250             12  
14        1360             13  
15        1465             14  
16        1510             14  
17        1580             15  
18        1600             16  
19        1660             18  
20        1740             18  
21        1785             19  
22        1885             20  
23        1880             21  
24        1980             23
```

Running priority queue-based Dijkstra's algorithm..  
Priority Queue-based Dijkstra's algorithm for List took 45 microseconds.  
Results for the Priority Queue-based Dijkstra's Algorithm:

Vertex	Distance from Source	Predecessor
0	0	-1
1	320	0
2	450	1
3	500	1
4	570	2
5	630	4
6	610	4
7	680	3
8	800	7
9	860	7
10	1000	9
11	1100	10
12	1130	10
13	1250	12
14	1360	13
15	1465	14
16	1510	14
17	1580	15
18	1600	16
19	1660	18
20	1740	18
21	1785	19
22	1885	20
23	1880	21
24	1980	23

Comparing results of both implementations:  
Priority Queue-based Dijkstra's algorithm is faster.

Both implementations produced the same results.