****

**Design and Analysis of Algorithms Project**

**Group Members:**

**Group Member 1:**

**Name:** Jiyad Khan

**Roll Number:** 19I-1771

**Group Member 2:**

**Name:** Ali Hussain

**Roll Number:** 19I-1701

**Group Member 3:**

**Name:** ArsalanAhmed

**Roll Number:** 19I-1875

**Section:** BS SE (Q)

**Group:** H

**Machine Specification:**

**Group Member 1:**

**Platform:** Windows

**Platform Version:** 10.0.19041

**Hostname:** DESKTOP-HJ1B6DL

**IP Address:** 192.168.56.1

**Processor:** Intel64

**RAM:** 8 GB

**Group Member 2:**

**Platform:** Windows

**Platform Version:** 10.0.19041

**Hostname:** DESKTOP-HJ1B6DL

**IP Address:** 192.168.56.1

**Processor:** Intel64

**RAM:** 8 GB

**Group Member 3:**

**Platform:** Windows

**Platform Version:** 10.0.19041

**Hostname:** DESKTOP-HJ1B6DL

**IP Address:** 192.168.56.1

**Processor:** Intel64

**RAM:** 8 GB

**Details of the Dataset:**

**Name of Dataset:** Google Web Graph

**Link for Dataset:** <http://snap.stanford.edu/data/web-Google.html>

**Details:** Nodes represent web pages and directed edges represent hyperlinks between them. The data was released in 2002 by Google as a part of Google Programming Contest.

**Data Statistics:**

**Nodes:** 875713

**Edges:** 5105039

**Nodes in largest WCC:** 855802

**Edges in largest WCC:** 5066842

**Nodes in largest SCC:** 434818

**Edges in largest SCC:** 3419124

**Average Clustering Coefficient:** 0.5143

**Number of triangles:** 13391903

**Fraction of closed triangles:** 0.01911

**Diameter:** 21

**Assumption:** As our graph wasn’t weighted, to solve weight involving algorithms we have assigned random weights to our dataset.

**Complexity Analysis of all Algorithms:**

**Time complexity of BFS:**

The time complexity of DFS is O(V+E), where E = No. of edges & V = No. of Vertices

**Time complexity of DFS:**

The time complexity of DFS is O(V+E), where E = No. of edges & V = No. of Vertices

**Time complexity of Kruskal’s Algorithm:**

The time complexity in Kruskal's algorithm is O(E log V), where V is the number of vertices.

**Time complexity of Prim’s Algorithm:**

Prim's algorithm has a time complexity of O(V^2), V being the number of vertices.

**Time complexity of Dijkstra’s Algorithm:**

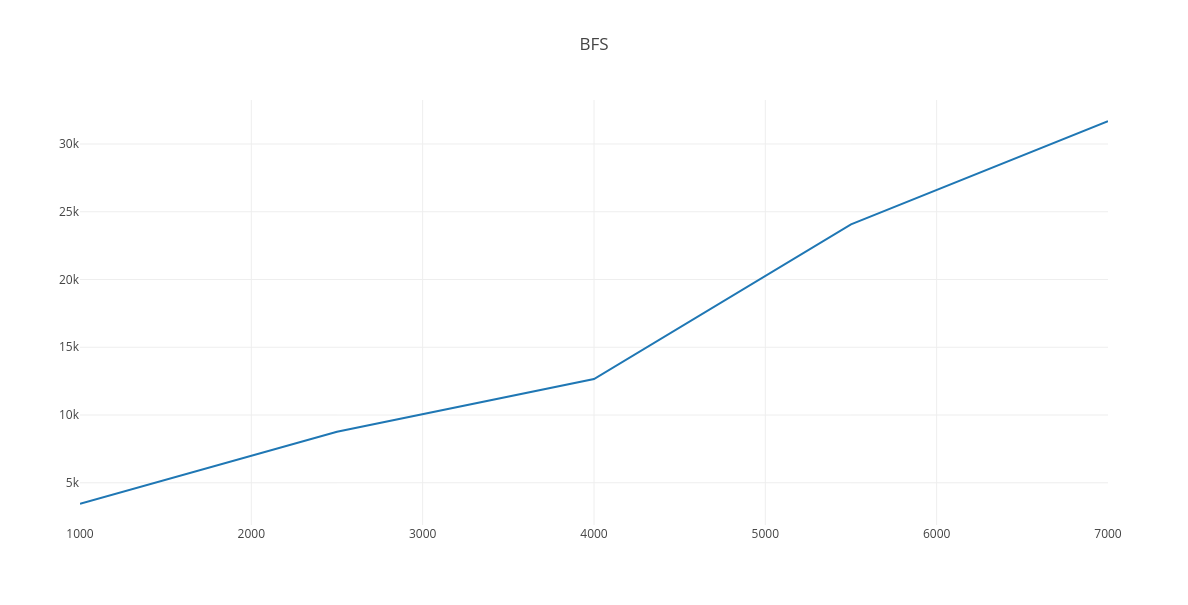
Time complexity of Dijkstra's algorithm is O(E log V), where V is the number of vertices in the graph.

**Time complexity of Bellman Ford’s Algorithm:**

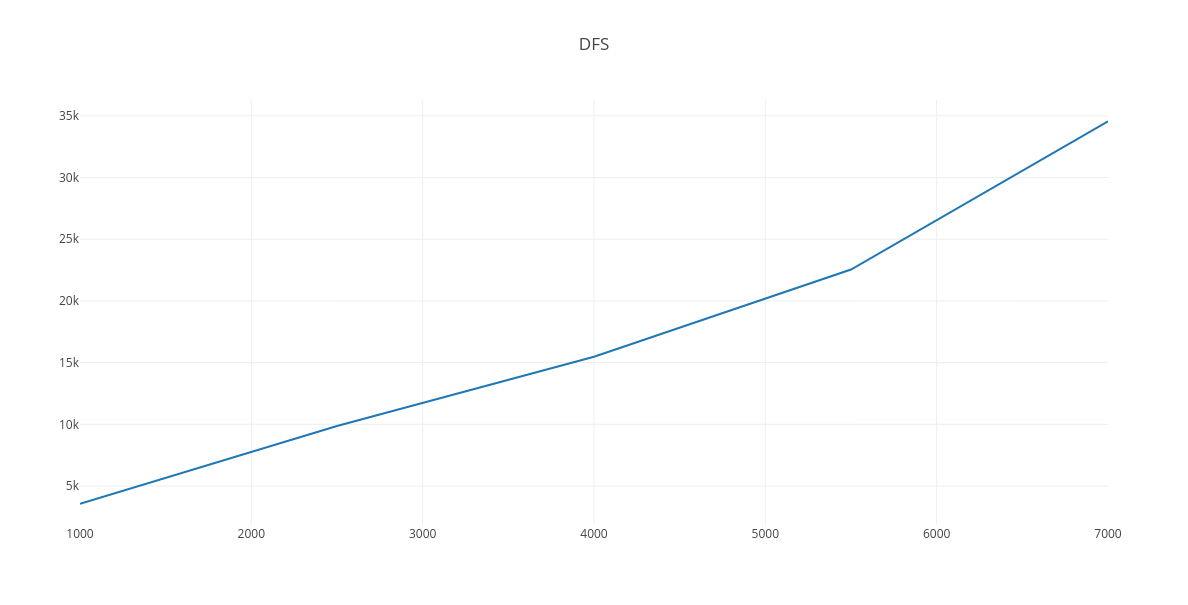
Time Complexity of Bellman Ford algorithm is relatively high O (V \* E).

**Performance Graph:**

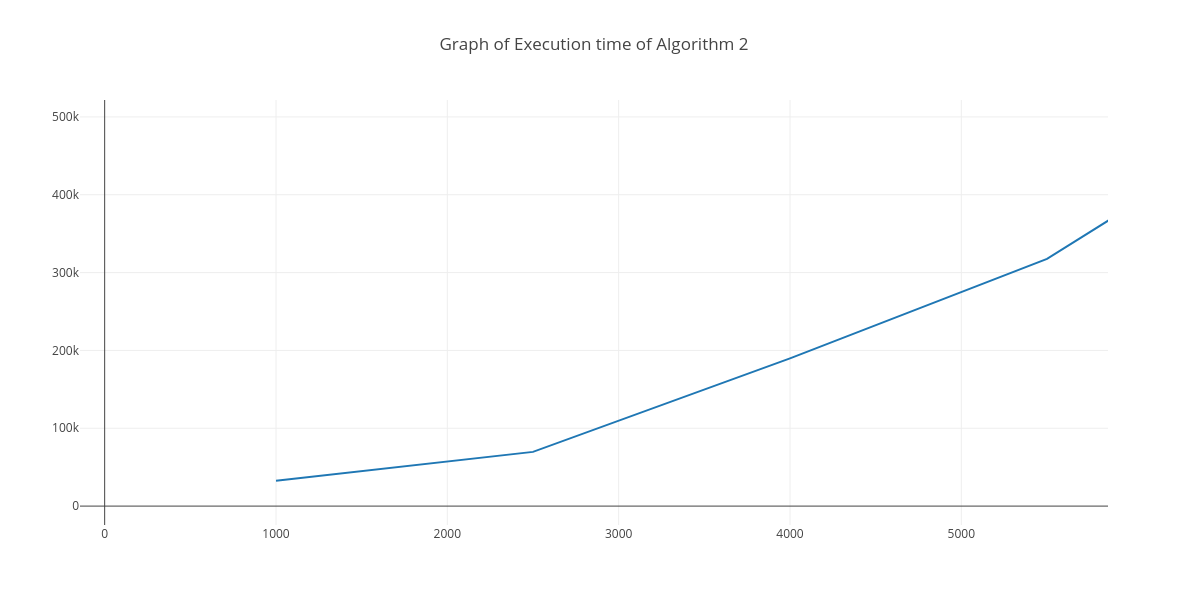
**BFS**



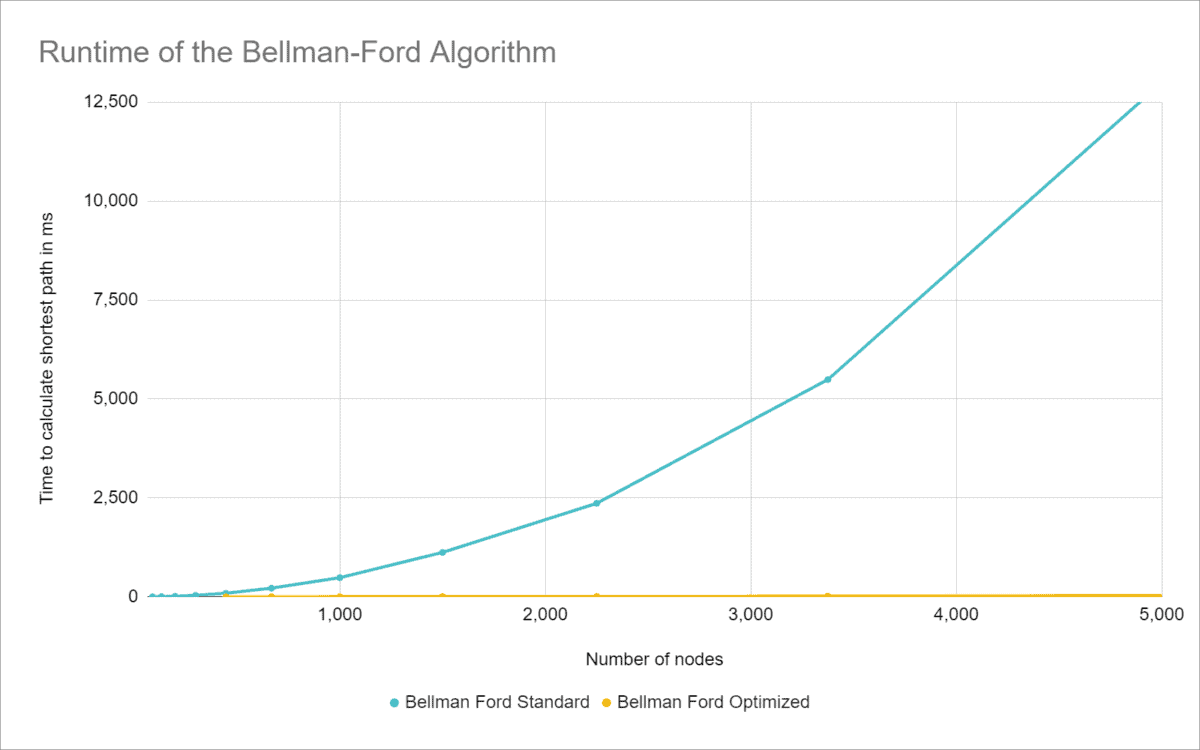
**DFS**



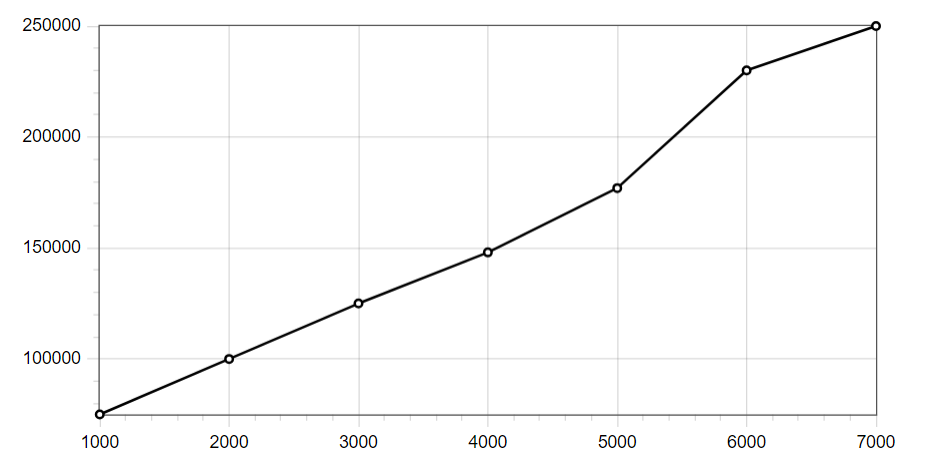
**Dijkstra**



**Bellman Ford**



**Kruskal**



**Prim**

