**Documentation**

-----2022 Spring Semester

-----Submission Date: Feb.27th.2022

-----**Peilin He,** [**peh53@pitt.edu**](mailto:peh53@pitt.edu)**, 4447252**

PS: This project were tested by two different computers. The first two Dijkstra’s algorithms were executed by Intel Core i7 7700HQ, 3.6 GHZ freq. The last two Floyd’s algorithms were executed by Apple M1 Max. 3.2 GHZ freq. So, the data could be somehow different, but the trends are the same.

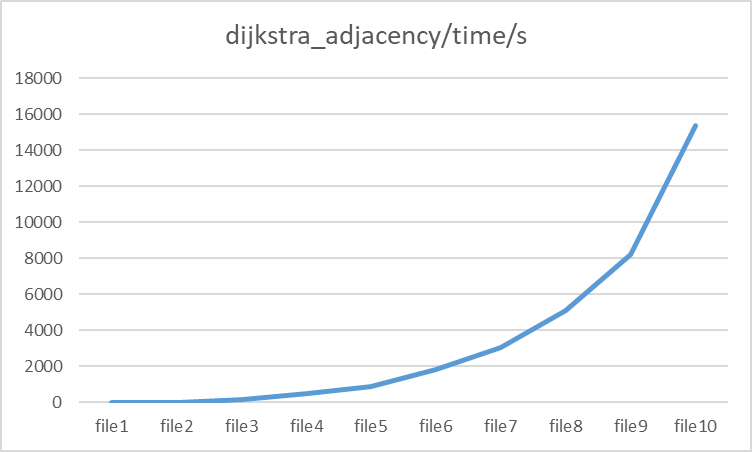
**1. Dijkstra’s algorithm with adjacency matrix.**

**Result:**

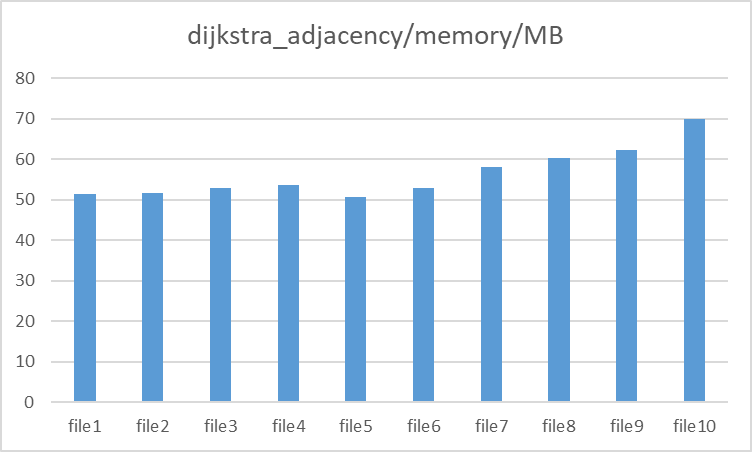
文本

描述已自动生成

**Time plot:**

****

**Memory plot:**

****

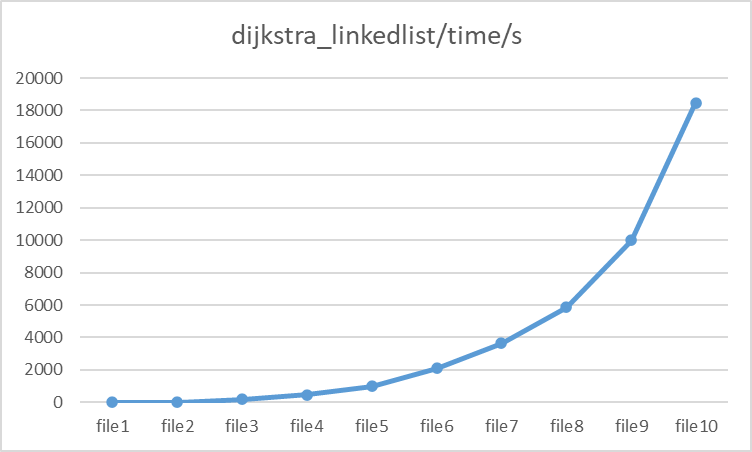
**2. Dijkstra’s algorithm with linked list.**

**Result:**

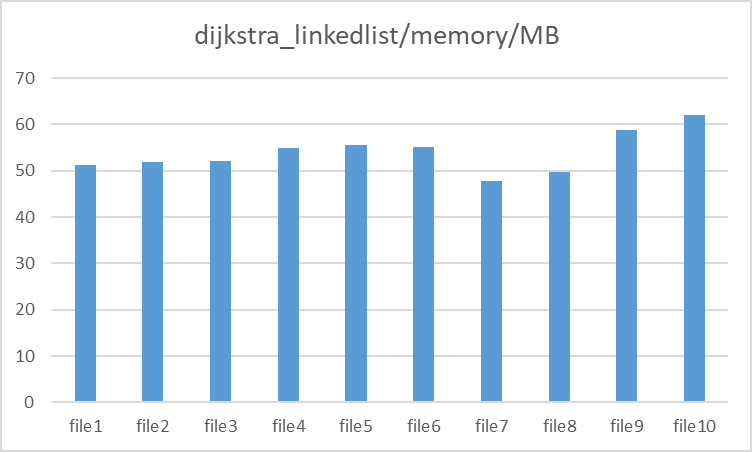
文本

描述已自动生成

**Time plot:**

****

**Memory plot:**

****

**3. Floyd’s algorithm with adjacency matrix.**

**Result:**

**Text

Description automatically generated**

**Time plot:**

Chart, line chart

Description automatically generated

**Memory plot:**

Chart, line chart, scatter chart

Description automatically generated

**4. Floyd’s algorithm with linked list.**

**Result:**

**Text

Description automatically generated**

**Time plot:**

Chart, line chart

Description automatically generated

**Memory plot:**

Chart, line chart, scatter chart

Description automatically generated

**Summary:**

**Time analysis:**

For the time consumption, the Floyd’s algorithms are more efficient than Dijkstra’s algorithms; adjacency matrix algorithms are efficient than linked list algorithms. Taking these four algorithms in order, Floyd’s adjacency matrix > Floyd’s linked list > Dijkstra’s adjacency matrix > Dijkstra’s linked list. The space complexity of these 4 algorithms is equal to O(n2). The time complexity of these 4 algorithms is equal to O(n\*logn).

**Memory analysis:**

For the RAM usage, linked list algorithms are better than adjacency matrix algorithms. The efficiency order is as follows: Dijkstra’s linked list > Floyd’s linked list > Dijkstra’s adjacency matrix > Floyd’s adjacency matrix.

In conclusion, Floyd’s algorithm is much faster than Dijkstra’s algorithms in practice. It computed all-pair shortest paths after running it once, however, Dijkstra’s algorithm must be repeated for different start nodes. The memory consumption vice versa.