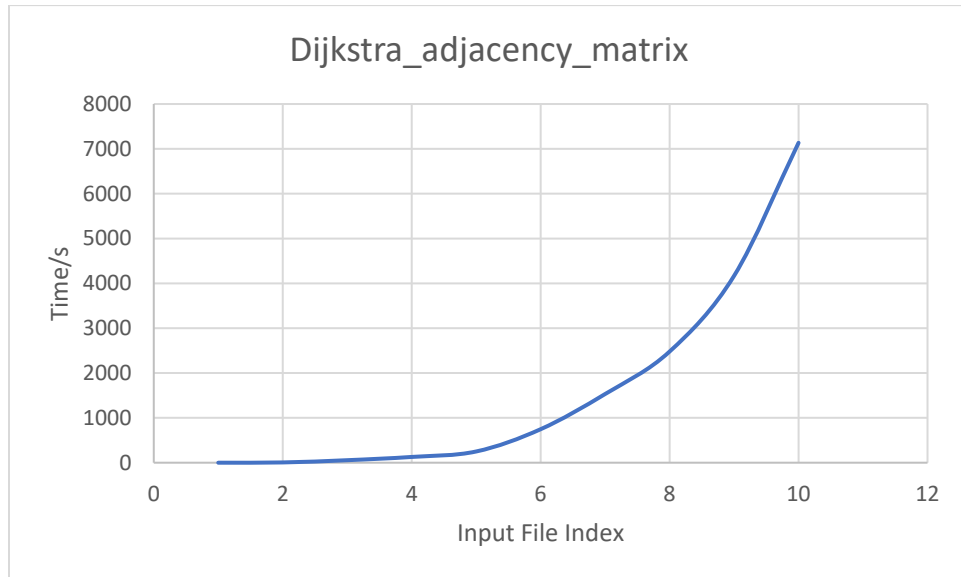


The result of Dijkstra's algorithm by adjacency matrix:

```
From 197 to 27, path is [197, 198, 303, 293, 142, 26, 27]  
From 65 to 280, path is [65, 216, 116, 117, 201, 274, 326, 24, 23, 125, 140, 203, 167, 197, 192, 280]  
From 187 to 68, path is [187, 238, 229, 231, 264, 247, 17, 18, 242, 158, 77, 78, 136, 137, 332, 70, 134, 176, 269, 286, 300, 318, 290, 302, 323, 277, 175, 68]
```

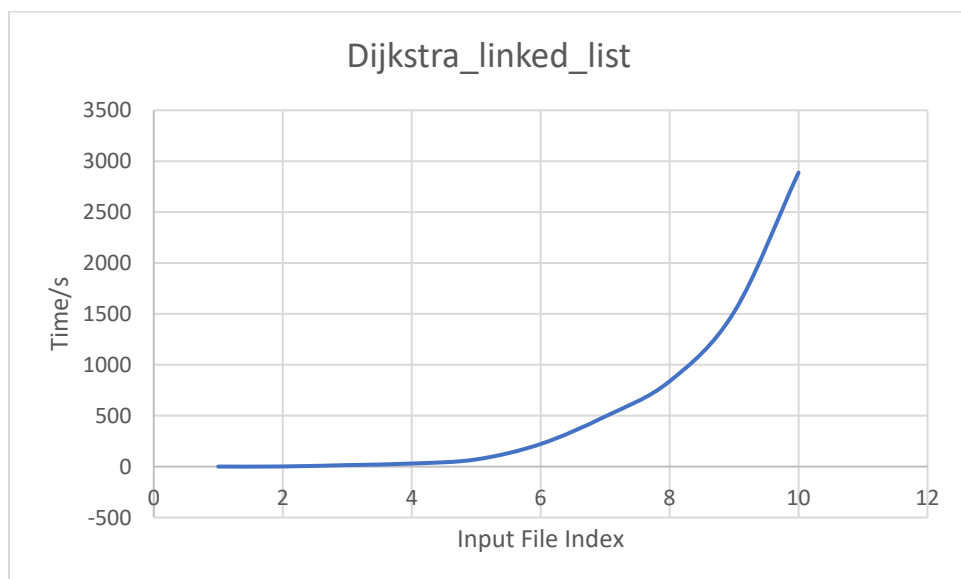
Plot of running time for each input file:



The result of Dijkstra's algorithm by linked list:

```
From 197 to 27, path is [197, 198, 303, 293, 142, 26, 27]  
From 65 to 280, path is [65, 216, 116, 117, 201, 274, 326, 24, 23, 125, 140, 203, 167, 197, 192, 280]  
From 187 to 68, path is [187, 238, 229, 231, 264, 247, 17, 18, 242, 158, 77, 78, 136, 137, 332, 70, 134, 176, 269, 286, 300, 318, 290, 302, 323, 277, 175, 68]
```

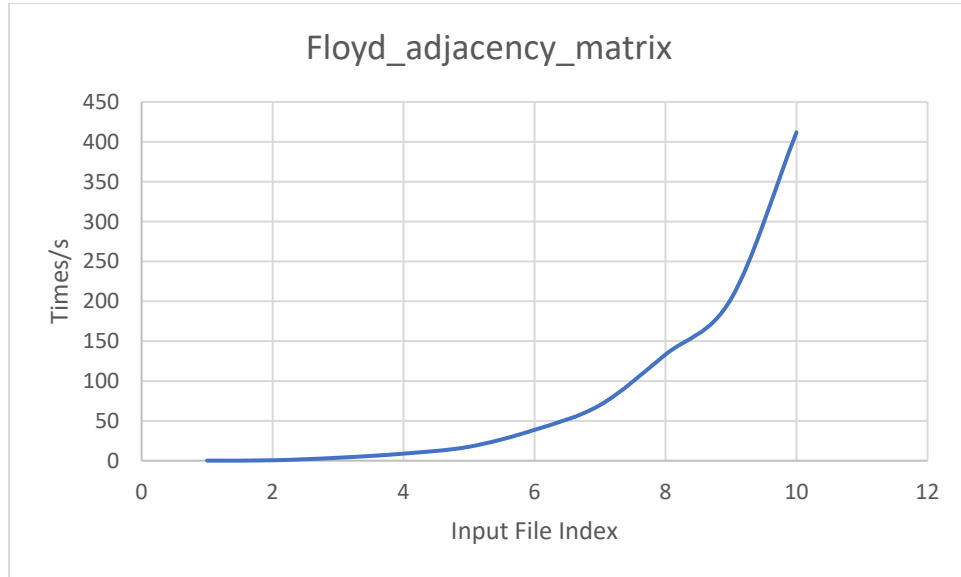
Plot of running time for each input file:



The result of Floyd's algorithm by adjacency matrix:

```
From 197 to 27, path is 197>>198>>303>>293>>142>>26>>27  
From 65 to 280, path is 65>>216>>116>>117>>201>>274>>326>>24>>23>>125>>140>>203>>167>>197>>192>>280  
From 187 to 68, path is 187>>238>>229>>231>>264>>247>>17>>18>>242>>158>>77>>78>>136>>137>>332>>70>>134>>176>>269>>286>>300>>318>>290>>302>>323>>277>>175>>68
```

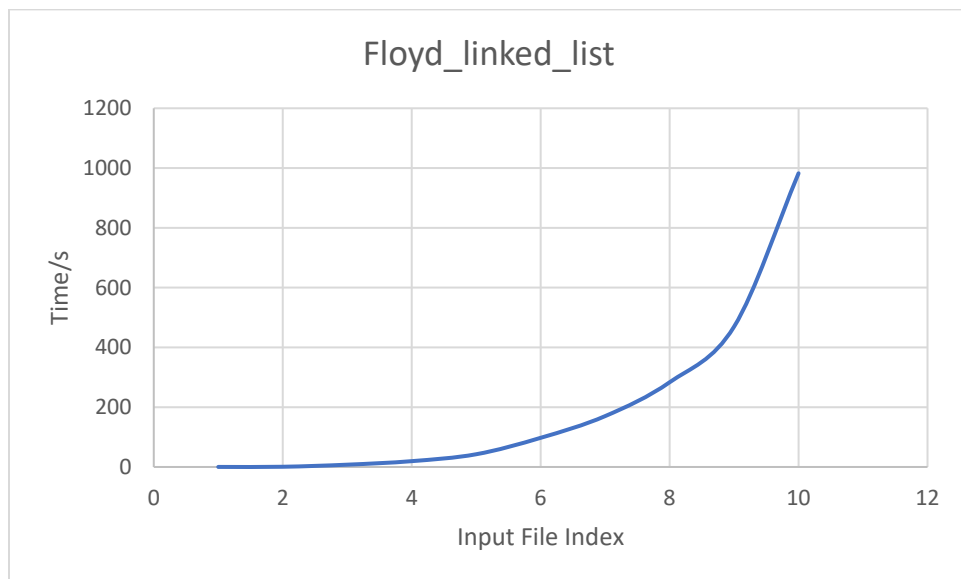
Plot of running time for each input file:



The result of Floyd's algorithm by linked list:

```
From 197 to 27, path is 197>>198>>303>>293>>142>>26>>27  
From 65 to 280, path is 65>>216>>116>>117>>201>>274>>326>>24>>23>>125>>140>>203>>167>>197>>192>>280  
From 187 to 68, path is 187>>238>>229>>231>>264>>247>>17>>18>>242>>158>>77>>78>>136>>137>>332>>70>>134>>176>>269>>286>>300>>318>>290>>302>>323>>277>>175>>68
```

Plot of running time for each input file:



Memory usage discussion:

Due to the poor processing ability of Java for computing memory space, I can't directly and intuitively show the program memory usage. I can only discuss it by analyzing the space occupied by data structure.

Dijkstra's Algorithm

Because of the Java, space store is difficult to calculate. For adjacent matrix, the memory usage of the adjacent matrix is $N \times N$, whose N is the number of all nodes. And the memory usage of paths is $N \times \text{path}$. For linked-list, the general memory usage is $(N \times N - N) / 2$, which will be $N \times (N - 1)$ when all path need to be completed.

Floyd Algorithm

For adjacent matrix, the memory usage of adjacent matrix, cost and path are all $N \times N$. For linked-list, the general memory usage is $(N \times N - N) / 2$, which will be $N \times (N - 1)$ when all path need to be completed.