Xinyu Chen
GTID: 903612905

A brief explanation of how I developed my program

Highlight some key points in my programs:

- Define helper function:
  - **newVertex**: allocate memory for new vertex and initialize it;
  - **relax**: relax algorithm, which is used both in Bellman-Ford and Dijkstra's algorithms;
  - **printMaximumPath**: generate and print the whole maximum-paths, which is used in findPaths function;
- In FindPaths function:
  - Use distance (array) to record the maximum length from the source vertex for each vertex
  - Use predecessor (array) to record the predecessor in maximum length for each vertex
  - set_Q is an array used in Dijkstra's algorithm. It stores the distance of vertices. If a vertex is not in the Q set, its' value will be INF.
  - Use for loop to find the minimum value in set_Q

Most import sample output

- I verified program with different arguments

Using default arguments

```
(base) xinyuchen@XyGray ~/workspace/CSE6010-22Fall/Assignments/A3 master ± make
clang -o divisorgraph main.c divisorgraph.c -O0 -g -Wall -Werror -std=gnu99
(base) xinyuchen@XyGray ~/workspace/CSE6010-22Fall/Assignments/A3 master ± ./divisorgraph
The number of vertices in graph is 1000, the search algorithm used is Dijkstra's algorithm
The length of maximum-length path in this divisor graph is 9.
These paths are:
1 -> 2 -> 4 -> 8 -> 16 -> 32 -> 64 -> 128 -> 256 -> 512
1 -> 2 -> 4 -> 8 -> 16 -> 32 -> 64 -> 128 -> 256 -> 768
```

Using 1 argument with number of vertices

```
(base) xinyuchen@XyGray ~/workspace/CSE6010-22Fall/Assignments/A3 master ± make
clang -o divisorgraph main.c divisorgraph.c -O0 -g -Wall -Werror -std=gnu99
(base) xinyuchen@XyGray ~/workspace/CSE6010-22Fall/Assignments/A3 master ± ./divisorgraph 7
The number of vertices in graph is 7, the search algorithm used is Dijkstra's algorithm
The length of maximum-length path in this divisor graph is 2.
These paths are:
1 -> 2 -> 4
1 -> 2 -> 6
```

Using 1 argument with algorithm

```
(base) xinyuchen@XyGray ~/workspace/CSE6010-22Fall/Assignments/A3 master ± make
clang -o divisorgraph main.c divisorgraph.c -O0 -g -Wall -Werror -std=gnu99
(base) xinyuchen@XyGray ~/workspace/CSE6010-22Fall/Assignments/A3 master ± ./divisorgraph b
The number of vertices in graph is 1000, the search algorithm used is Bellman-Ford algorithm
The length of maximum-length path in this divisor graph is 9.
These paths are:
1 -> 2 -> 4 -> 8 -> 16 -> 32 -> 64 -> 128 -> 256 -> 512
1 -> 2 -> 4 -> 8 -> 16 -> 32 -> 64 -> 128 -> 256 -> 768
```

Using 2 argumets

```
(base) xinyuchen@XyGray ~/workspace/CSE6010-22Fall/Assignments/A3 master ± make
clang -o divisorgraph main.c divisorgraph.c -O0 -g -Wall -Werror -std=gnu99
(base) xinyuchen@XyGray ~/workspace/CSE6010-22Fall/Assignments/A3 master ± ./divisorgraph 20 b
The number of vertices in graph is 20, the search algorithm used is Bellman-Ford algorithm
The length of maximum-length path in this divisor graph is 4.
These paths are:
1 -> 2 -> 4 -> 8 -> 16
```

What I found useful about the peer review process

- Add more comments to make the program easier to understand
- Fix bugs and now they are bug-free
- Using ‘atoi’ function to fix issues in command-line arguments
- Change the name of the variable to make it more clear