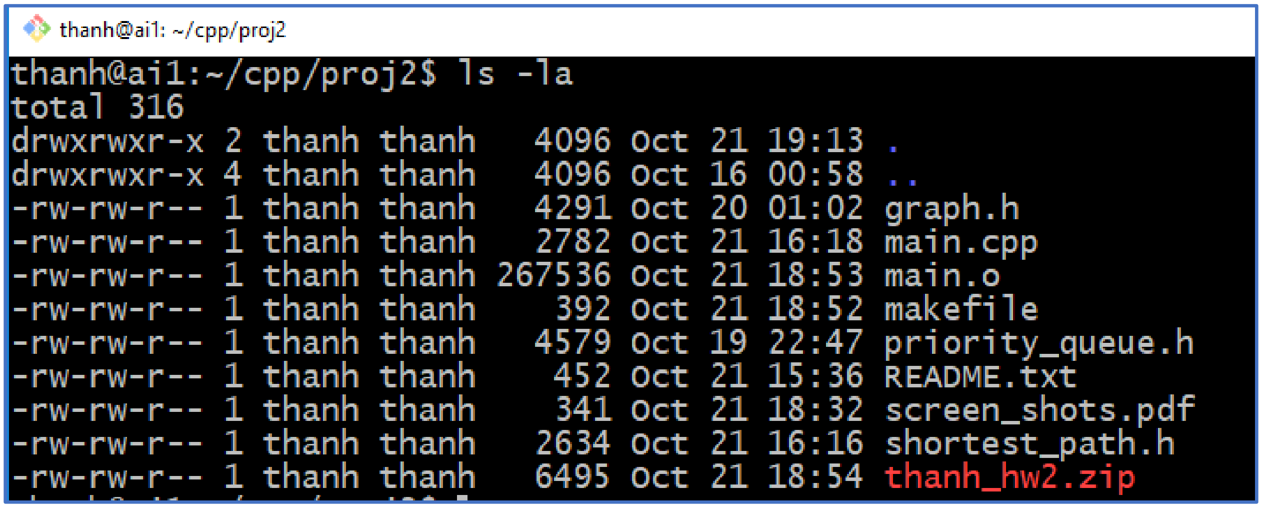
Compute shortest path average of random graphs with different edge densities

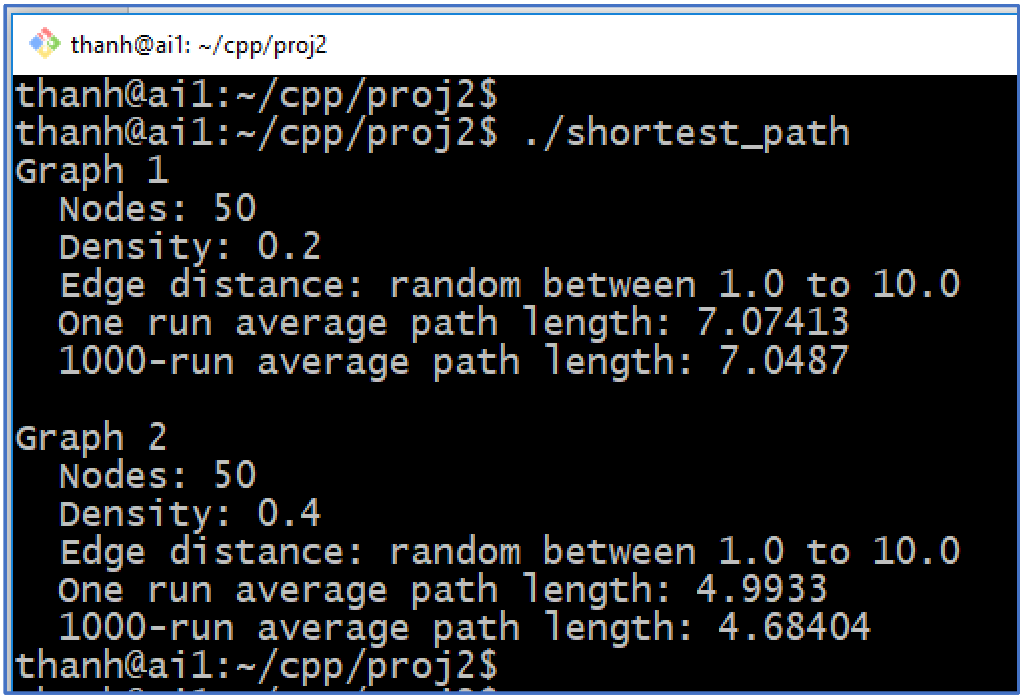
Screenshots of homework assignment #2

Screenshot #1 – list of submitted files inside thanh_hw2.zip



```
thanh@ai1:~/cpp/proj2$ ls -la
total 316
drwxrwxr-x 2 thanh thanh   4096 Oct 21 19:13 .
drwxrwxr-x 4 thanh thanh   4096 Oct 16 00:58 ..
-rw-rw-r-- 1 thanh thanh   4291 Oct 20 01:02 graph.h
-rw-rw-r-- 1 thanh thanh   2782 Oct 21 16:18 main.cpp
-rw-rw-r-- 1 thanh thanh 267536 Oct 21 18:53 main.o
-rw-rw-r-- 1 thanh thanh    392 Oct 21 18:52 makefile
-rw-rw-r-- 1 thanh thanh   4579 Oct 19 22:47 priority_queue.h
-rw-rw-r-- 1 thanh thanh    452 Oct 21 15:36 README.txt
-rw-rw-r-- 1 thanh thanh    341 Oct 21 18:32 screen_shots.pdf
-rw-rw-r-- 1 thanh thanh   2634 Oct 21 16:16 shortest_path.h
-rw-rw-r-- 1 thanh thanh   6495 Oct 21 18:54 thanh_hw2.zip
```

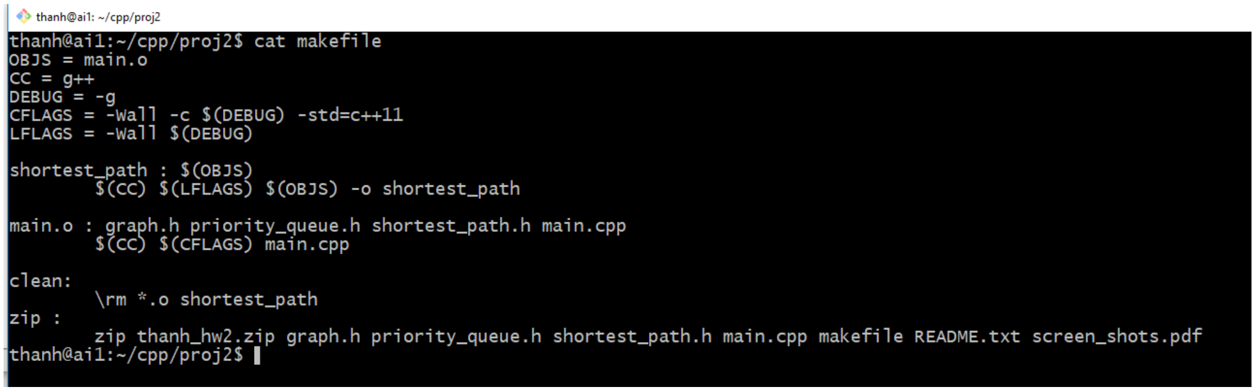
Screenshot #2 – output of the program on random graphs with 50 vertices, edge density 0.20 and 0.40 and random edge distance in the range [1.0, 10.0]



```
thanh@ai1:~/cpp/proj2$
thanh@ai1:~/cpp/proj2$ ./shortest_path
Graph 1
  Nodes: 50
  Density: 0.2
  Edge distance: random between 1.0 to 10.0
  One run average path length: 7.07413
  1000-run average path length: 7.0487

Graph 2
  Nodes: 50
  Density: 0.4
  Edge distance: random between 1.0 to 10.0
  One run average path length: 4.9933
  1000-run average path length: 4.68404
thanh@ai1:~/cpp/proj2$
```

Appendix A - How to run compile and run the program



```
thanh@ai1:~/cpp/proj2$ cat makefile
OBJS = main.o
CC = g++
DEBUG = -g
CFLAGS = -Wall -c $(DEBUG) -std=c++11
LFLAGS = -Wall $(DEBUG)

shortest_path : $(OBJS)
        $(CC) $(LFLAGS) $(OBJS) -o shortest_path

main.o : graph.h priority_queue.h shortest_path.h main.cpp
        $(CC) $(CFLAGS) main.cpp

clean:
        \rm *.o shortest_path
zip :
        zip thanh_hw2.zip graph.h priority_queue.h shortest_path.h main.cpp makefile README.txt screen_shots.pdf
thanh@ai1:~/cpp/proj2$ ▌
```

1. Create a new directory on your Linux (e.g. Ubuntu) machine
2. Unzip **thanh_hw2.zip** into the new directory created on above step
3. Type **make** and press ENTER to compile using gcc / g++ compiler
4. Type **./shortest_path** and press ENTER to run

- The program is compiled using the makefile as seen on the above screenshot
- The makefile uses standard GNU gcc/g++ compiler that comes by default as part of Ubuntu 16.04 Linux Operating System