

CSE 373 Homework 5 Write up

Kanghui Liu

Fangzheng Sun (writing)

Who is in your group (Give name, UW NetID & student number of each person)?

- 1) **Describe the worst-case asymptotic running times of your methods `adjacentVertices`, `edgeCost`, and `shortestPath`. In your answers, use $|E|$ for the number of edges and $|V|$ for the number of vertices. *Explain and justify your answers.***

The worst-case runtime of `adjacentVertices` is $O(|V|)$.

To return the collection of the vertices adjacent to the given vertex, we have to firstly extract the collection of the edges with the keyset to be the given vertex, and then use a for-each loop to put all destinations to the collection. The worst case is that the given vertex is adjacent to all other ones. So the run time will be $V - 1$, which is $O(|V|)$.

The worst-case runtime of `edgeCost` is $O(|V|)$.

To return the cost between a and b, we have to find the corresponding edges of a in the map, and then using a for-each loop to check all of them until there appears the vertex b. So the worst case is that there's $V - 1$ items in the edges corresponding to a, and the last element to check is b. The runtime is $O(|V|)$.

The worst-case runtime of `shortestPath` is $O(|V|^2)$.

This method is a little bit long. So we calculate it by parts. First, copying the whole vertices collection needs $O(|V|)$. Then the big for loop's runtime is $O(|V|^2)$.

So the run time is $O(|V|^2)$.

- 2) **Describe how you tested your code.**

In `TestGraph.java`, I create a new graph with 7 vertices named A, B, C, D, E, F, G. The cost of two positively adjacent characters, such as A to B (or B to A), is 1, regardless positive or negative. If there is a path not between adjacent characters, the cost will be 10 more times the distance. For instance, the cost between A to C (or C to A) will be 20. Thus, obviously the shortest path between two characters will be the series of all characters between them, including the two characters. The two files are called `alpha.txt` and `cost.txt`. The codes are based on `FindPaths.java`, and the algorithm is basically the same except that after typing the character, there will be some testing code to compute whether the graph really gives the optimal solution.

- 3) **If you worked with a partner:**
a) **Describe the process you used for developing and testing**

your code. If you divided it, describe that. If you did everything together, describe the actual process used (eg. how long you talked about what, what order you wrote and tested, and how long it took).

We divided the work. We read the homework specification and worked for one afternoon to divide it.

I firstly wrote Edge.java and findPaths.java. Liu wrote Vertex.java and MyGraph.java. After she finish her part, I wrote TestGraph.java to check whether the previous four files have something wrong. This part took us about a weekend.

Then we work together to adjust the errors. We gathered for one afternoon to do this.

Finally I did the write-up questions and came up with the first one in above and beyond. And Liu did the rest of the extra-credit part. This took as about one day.

b) Describe each group member's contributions/responsibilities in the project.

For Kanghui Liu:

Writing Vertex.java and MyGraph.java

Run the code until it succeed based on the finished 4 java files.

Do the 2, 3 questions of extra credit.

For me (Fangzheng Sun):

Writing Edge.java and findPaths.java

Testing the code and writing TestGraph.java

Writing readme.pdf

First question in above-and-beyond

c) Describe at least one good thing and one bad thing about the process of working together.

Good thing: Working together makes the long homework seems shorter since each individual will have his or her part instead of the whole. And we can talk with each other if we meet difficulties.

Bad thing: To understand what other group members' code will take me some time. Sometimes I even have to rewrite the code to fully understand it.

4) If you did any above-and-beyond, describe what you did.

For the first question in above-and-beyond, I found registration of courses is really close to the graph. I am in ACMS major. In my option, I believe that Amath 403 is the most difficult course. The prerequisite is Amath 401 and math 307. But if I took math 309 first, this course will be much easier. The same problem also appear for Amath 401 or some courses else. Assume I learn one math course each quarter and can stay here as many quarters as I want. So let us make a table of

relevant courses, and how many hours it takes me to pass the class if I study some other courses in the last quarter. My terminal is to successfully pass the course Amath 403. Everything starts from nothing, assuming I have no AP scores.

Last quarter course	This quarter course	Time he needs to pass this course (hours)
NOTHING	MATH124	80
MATH124	MATH125	100
MATH125	MATH126	150
MATH126	MATH307	80
MATH307	MATH324	80
MATH324	AMATH401	100
MATH307	AMATH401	300
MATH307	MATH309	100
MATH309	MATH324	80
MATH309	AMATH401	200
MATH309	AMATH403	300
AMATH401	AMATH403	100

These may not be true statistics, we just assume so. Then we write them in two files called courses.txt and hours.txt and use the code we write to compute the shortest time and path to study from NOTHING to AMATH403.

The result of running FindPaths.java:

```

Start vertex? NOTHING
Destination vertex? AMATH403
The shortest path from NOTHING to AMATH403 is:
NOTHING MATH124 MATH125 MATH126 MATH307 MATH324
AMATH401 AMATH403
The cost is 690

```

I previously assume that taking MATH 309 will make things easier. Surprisingly, the shortest path shows no MATH 309 at all. And it takes me at least 690 hours to pass AMATH 403 from the beginning.

Appendix

Place anything that you want to add here.