AP Computer Science
Dr. Paul L. Bailey

Responses 03/30
Sunday, March 29, 2020

Question 1. For caesar key, are we supposed to square the difference or take the square root of the difference of the frequencies? Also, how do we compare the frequencies of the decrypted strings to the frequency of the actual string?

Answer. Since we are using "distances" only for comparison, it is unnecessary to take the square root, since if a and b positive, $a < b$ if and only if $a^2 < b^2$.

For each possible key, encrypt the (already encrypted) string, get the frequency of the result, and compare it to the frequency of standard english. The closet key you find is the decryption key. The original encryption key is the inverse key of the decryption key. □

Question 2. I'm unsure how you want the print method for each class to be done especially the graph class. For Graph, we do need two private Arraylists/Lists right? Moreover, I'm unsure what we're supposed to do for `Graph.add(Vertex v, List<Vertex> L)`- could you clarify on this? Lastly, what are we returning in Edge.hashCode()?

Answer. I am asking you to think about how to construct the classes, and you have. I think two arraylists would work well.

Print the graph in a way that allows you to see everything you need to see. I would probably list the vertices as compactly as possible, then list the edges.

I guess `Graph.add(Vertex v, List<Vertex> L)` is a little obscure. In my code, it adds v to L only if v is not already in L.

When you override the equals method, it is "best practice" also override the hashcode method. The only rule is, if two things are "equal" according to the equals method, they must have the same hashcode. Some of the library methods use hashcodes for efficiency. In this case, returning the number of vertices might work well enough. □

Question 3. I was reading PCScix35aWas2016, and I am confused about the definition of a Eulerian path. Wouldn't all connected graphs exhibit the same number of Eulerian paths as their vertices?

Answer. That is a good question. I will respond based on the definitions given in Was2016.

Yes, an Eulerian *trail* exists for every connected graph. But for some connected graphs, it is not possible to traverse every edge without passing through a nonterminal vertex more than once. This is the famous *Koenigsberg bridge problem*.

`https://www.mathsisfun.com/activity/seven-bridges-konigsberg.html`

□