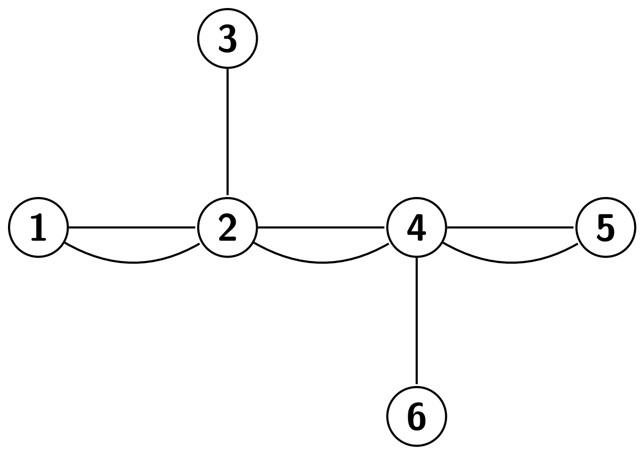
On finding the longest Eulerian path with multiple edges

1 Introduction

In this report I am going to propose an algorithm that allows to determine the length of the longest path through vertices that have the same number of multiple edges and provide an example on the application of the proposed algorithm.
To begin with, consider the following graph.



From this multigraph (Def:graph containing multiple edges) it is quite clear by mere observation, that the the longest path through the vertices with two double edges is of distance 3 (if we give weight 0 to each vertex and 1 to edges the desired multiple edges if the treat all of them as one. This assumption is going to be used for the whole report).
Now consider the adjacency matrix of this graph, which is given as

$$\begin{pmatrix} 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

From the adjacency matrix, as expected, we observe that there are 3 double edges, as there are 6 entries, but we know that the graph is connected and the matrix is symmetric. But what about arbitrary big graphs? Plotting and counting is clearly not a way to solve the problem, therefore we have to extract the information from the adjacency matrix for the graph.

2 Solution

Consider an N×N adjacency matrix for a finite, loopless , undirected (as we would have a problem with contribution of loops) multigraph with N verticies and with graph multiplicity of M (Def:Graph multiplicity is the maximum edge multiplicity. The edge multiplicity of a given vertex in a multigraph is the number of multiple edges sharing that vertex.). The adjacency matrix for such a graph would be given as

$$\begin{pmatrix} 0 & \lambda_1 & \lambda_2 & ... & ... & \lambda_n \\ \lambda_1 & ... & ... & ... & ... & ... \\ ... & ... & ... & M & ... & ... \\ ... & ... & M & ... & ... & ... \\ ... & ... & ... & ... & ... & ... \\ \lambda_n & ... & ... & ... & ... & 0 \end{pmatrix}$$

Where $\lambda_{1...n} \in \mathbb{N}$ and $\lambda_{1...n}$ states for edge multiplicity of a vertex with multiplicity $< M$.
One of the M values is in (k,j) and (j,k) positions, the rest is also somewhere in the matrix.

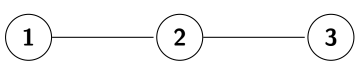
Now we observe that as we are interested in a path trough the vertices with the maximal edge multiplicity, we in fact can ignore all the values in the adjacency matrix that are $\neq M$.

Now we obtain the matrix that is given as

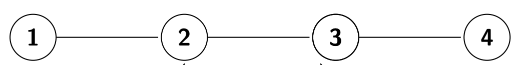
$$\begin{pmatrix} 0 & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... \\ ... & ... & ... & M & ... & ... \\ ... & ... & M & ... & ... & ... \\ ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & 0 \end{pmatrix}$$

Where we have only zero entries or M entries. We also observe that as a result of deletion of all the values that are not M we can result in a disconnected graph, but we are still interested in finding the longest path in the original graph, so we have to consider each component separately.

To do that we have to deduce how many and what are the components that we have. To understand how we can access this information, consider the following example, before introducing the general case. Consider the following two graphs and their adjacency matrices.



$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

And now consider disjoint union of the above graphs and consider that as a new graph. Its adjacency matrix is going to be as follows

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

It may be difficult to observe as there are a lot of zeros, but in fact the adjacency matrix of this graph in union of its components, but with components being separated with zero matrix of the size of the adjacency matrix of the component (and we know that the size is the number of vertices)

Therefore the claim is as follows:
Determining the components of a graph by an adjacency matrix
Consider an arbitrary finite, loopless, undirected, unconnected graph and its adjacency matrix with N vertices and M components.
Then in the adjacency matrix of size $N \times N$ for that graph the adjacency matrix will be a block-diagonal matrix with M square matrices of size corresponding to the number of verticies of that component.
Proof
From the adjacency matrix for an unconnected graph we observe that for a vertex after some (k,j) and (j,k) entry there will only zero entries, as one component will end and these vertices will no longer be connected to any other vertices. That will create a row/column of zeros after some (k,j) and (j,k) entry, but there still will be non-zero entries in the matrix, but more precisely, there will be an another square matrix that will represent an another matrix. For some graph with M component the adjacency matrix will be a block diagonal matrix, which is a square matrix with square matrices $A_1, ..., A_M$ lying along the diagonal and all the other entries of the matrix equal to 0.

Extra claim
Consider an arbitrary finite, loopless, undirected, connected graph with N vertices. Then for any partition of the original graph by edge-cut that results in formation of an unconnected graph with M components such that no component is an isolated vertex (vertices with deg(v)=0) are produced, the maximum number of components is $\leq \frac{N}{2}$

Proof
Consider an arbitrary finite, loopless, undirected, connected graph with N vertices. Then if one performs sufficient number of edge-cuts so that an unconnected graph is obtained, but no component is an isolated vertex, then the smallest component of such an unconnected graph consists of two vertices connected by an edge. Therefore if one performs a partition so that the maximal number of components is obtained, then the maximal number of components is $\frac{N}{2}$ for N even, and $\frac{N-1}{2}$ is N is odd.

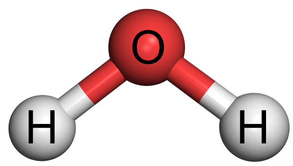
This is all the information that we need, now I shall state the overall process of finding such a maximal path.
Algorithm

1. For an arbitrary graph write its respective adjacency matrix.
2. To determine the maximal path through vertices with k-multiplicity, delete all the entries other than k from the adjacency matrix. Replace k by 1 for simplicity.
3. Analyze the connectivity of the obtained graph by adjacency matrix as stated above. If the graph is connected, go to step 5
4. If the graph is not connected, run breadth first search until there are no components.
5. For each component's adjacency matrix obtained, or if not obtained, for the whole matrix run the code in the appendix, which determines the eulerian path.
6. If the graph is not connected, the longest path with multiple edges for the original graph is the longest eulerian path of one of the components. If the graph is connected, the output is the longest path with multiple edges for the original graph.

Remark
If we initially given an acyclic graph, the procedure is quite nice and easy. We perform steps 1 and 2 from the list above, but now we can just take our adjacency matrix to the power of $N \in \mathbb{N}$ such that when taken to the power of $N + 1$ the adjacency matrix becomes zero matrix. Taking adjacency matrix to the power of N tells us how many paths there are that are of distance N, and when at $N + 1$ it becomes zero matrix, it means that there are no paths of distance $N + 1$, therefore the maximum length is N (check proposition 2.2 in the lecture notes for the proof and for the precise definition for the power of adjacency matrix).

Application
As was stated in the very beginning, I am going to utilize the proposed algorithm to assess a real world problem.
In chemistry there exists a term "conjugated system". This system is a system of connected p-orbitals also known as double bonds. For molecules that have different conformations and conjugated systems, the energy required to excite the electrons in a conjugated system is smaller for molecules that have longer conjugated system than for those having shorter conjugated system.
Or in very simple words, if a molecule has more double bonds one after another, less energy is required to do something with it.

Now consider a photo of a water molecule structure, two hydrogen (H) atoms, one oxygen (O) and two O-H bonds.



One can clearly see the resemblance with graphs. Therefore we can try to treat molecule's structure as a graph, denoting atom as a vertex, bond as an edge. With this convention now we shall consider two conformations of phenolphthalein (just a molecule with nice properties).

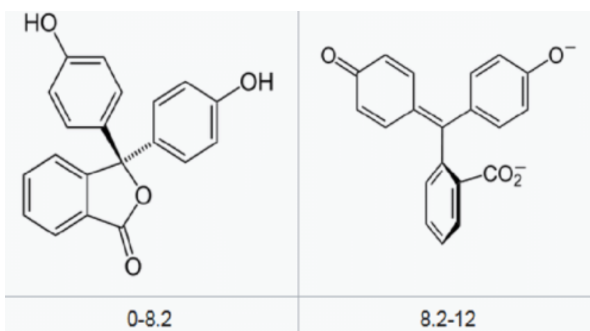


Figure 1: Two conformations of phenolphtalein

One clearly observes that for these conformations the length of the conjugated system (sequence of double edges/bonds) differs. Now we shall construct adjacency matrices for these graphs.

Although when I have obtained the matrices, they are apparently too big for LaTeX to write (they are 24×24). Therefore I have processed the data on my own using the algorithm stated above and the results are as follows.

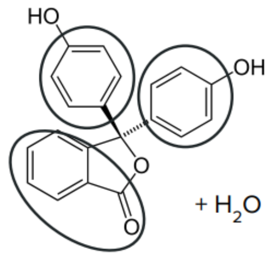


Figure 2: Phenolphtalein in pH 0-8.3

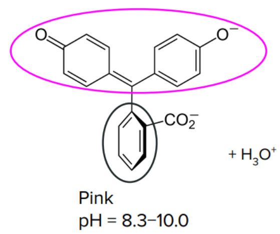


Figure 3: Phenolphtalein in pH 8.3-10

I have circled every conjugated system (chain of double bonds) in each conformation of the molecule. For figure 1 we observe that when step 2 is performed, we obtain 3 components (they are circled), and for figure 3 we obtain 2 components (also circled). One observes that in the second picture the chain in pink is the longest, and the eulerian path is of distance 10
Remarks

1. In the cycle in the molecule all the edges were considered as double, even if on the picture some edges are depicted as single. This is done to account for chemistry specific matter.

2. One may ask, why the distance is 10, if there is a single edge that connects two double edges in the pink circles? The answer is that due to the specifics of the definition of conjugated system, if there exists one single bond between two double bonds, then this single bond does not end the conjugated system, as conjugated system is an alternating sequence of single and multiple bonds. In the computation this remark was considered.

Appendix
The code utilized in the algorithm was taken from the following website (here, this is a hyperlink) or http://www.e-maxx-ru.1gb.ru/algo/euler(here add underscore)path and hereby the credit is given to a Russian guy who wrote this code when I was 3 years old. It really works, Jeffrey Zhang helped to fix the issues and the credit is given to Jeffrey Zhang for consulting me with graph theory questions.