Graphs and Their Applications (2)

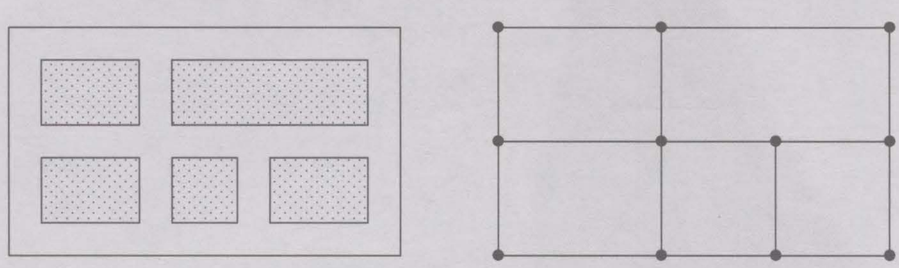
Koh Khee Meng
Department of Mathematics
National University of Singapore
Singapore 117543

Graphs and Their Applications (2)

Graphs and Their Applications (2)

5. Connectedness

The mathematical structure: graphs and, more generally, multigraphs, which was introduced in [3], can be used conveniently to model many real situations. For instance, Figure 5.1 (a) shows a section of the street system of a town, and it can be modeled as a graph as shown in Figure 5.1 (b), where vertices representing junctions of streets and two vertices are joined by an edge if and only if the corresponding junctions are linked by a street. For certain purposes, we may have to traverse the street system by passing through some junctions and streets. In order to show more precisely and succinctly the way we traverse, in this section, we shall introduce some basic terms in multigraph which serve the purpose.



(a) Figure 5.1 (b)

Consider the multigraph G of Figure 5.2 (that is, Figure 2.4 in [3]). If we start

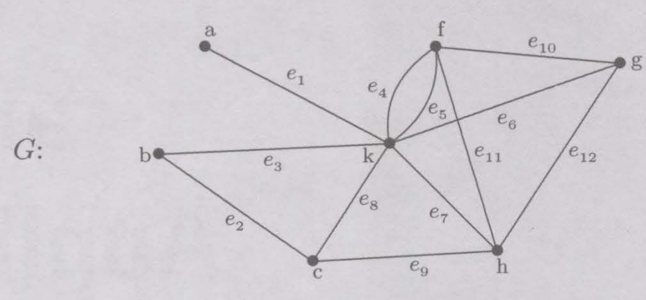


Figure 5.2

at vertex 'a', then we can reach vertex 'k' via the edge e_1, and from 'k' to 'h' via e_7. We can further proceed to reach 'g' via e_{12}. This process can be conveniently expressed by the following sequence of vertices and edges:

$$a\, e_1\, k\, e_7\, h\, e_{12}\, g. \qquad (1)$$

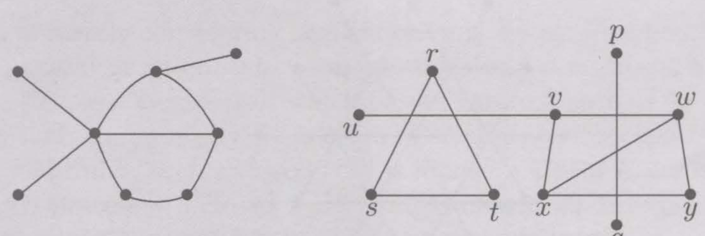
Such a sequence is called a *walk* or, more precisely, an $a-g$ *walk* as 'a' and 'g' are respectively the initial and terminal vertices of the walk. Note that the sequence

'$a\ e_1\ k\ e_8\ h$' is *not* a walk as the edge e_8 does not join the vertices 'k' and 'h'. Some more walks in G are given below:

$$be_3ke_4fe_5ke_4fe_{10}g, \quad (2)$$
$$be_3ke_4fe_5ke_7h, \quad (3)$$
$$be_3ke_4fe_{11}he_9c, \quad (4)$$
$$be_2ce_8ke_4fe_5ke_3b, \quad (5)$$
$$be_2ce_9he_{12}ge_6ke_3b. \quad (6)$$

While the definition of a 'walk' is quite general, in certain situations, we do need certain types of walks. A walk is called a *trail* if no edge in the walk is traversed more than once. A walk is called a *path* if no vertex in the walk is visited more than once. Thus, the $b-g$ walk (2) is not a trail; the $b-h$ walk (3) and $b-b$ walk (5) are trails but not paths; the $a-h$ walk (1) and $b-c$ walk (4) are paths. A walk is *closed* if its initial and terminal vertices are the same; and *open* otherwise. Thus, the walks (5) and (6) are closed while (1) - (4) are open. A closed walk is called a *cycle* if, besides the initial and terminal vertices (which are the same in this case), the rest are all distinct. Thus, the closed walk (6) is a cycle while the closed walk (5) is not. Note that the closed walk fe_4ke_5f is regarded as a cycle.

Any of the above notions of 'walks' enables us to introduce a very important class of multigraphs, called connected multigraphs. A multigraph G is said to be *connected* if every pair of vertices in G are joined by a path. For instance, in Figure 5.3, the graph (a) is connected while (b) is not so (observe that the vertices 'r' and 'u' are not joined by a path).



(a) (b)

Figure 5.3

A multigraph is *disconnected* if it is not connected. Observe that the disconnected graph (b) in Figure 5.3 is made up of three pieces that are themselves connected:

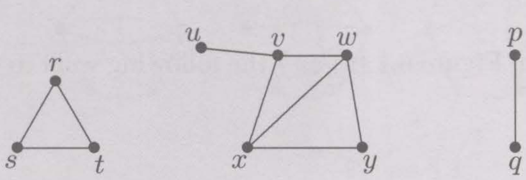


Figure 5.4

Each of these pieces is called a *component* of the graph (b).

Exercise 5.1. Consider the following graph

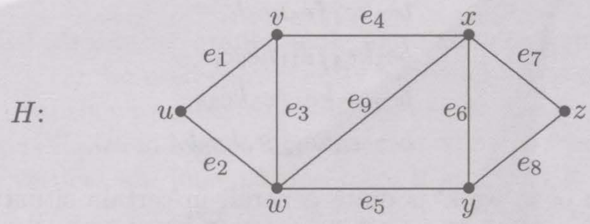


Figure 5.5

(a) Which of the following sequences represents a $u - z$ walk in H?
  (i) $ue_2we_5xe_7z$
  (ii) $ue_1ve_5ye_8z$
  (iii) $ue_1ve_3we_3ve_4xe_7z$

(b) Find a $u - z$ trail in H that is not a path.

(c) Find all $u - z$ paths in H which pass through e_9.

Exercise 5.2. Consider the following graph with 12 vertices and 9 edges. Is the graph connected? If not, how many components does it have?

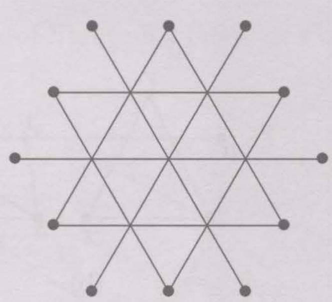


Figure 5.6

6. The Unicursal Property and Eulerian Multigraphs

Consider the multigraph G of Figure 6.1 (a) and the following walk in G as shown in Figure 6.1 (b):

$$W : xe_1we_4ye_5we_3xe_7ze_8ye_9ze_6we_2x$$

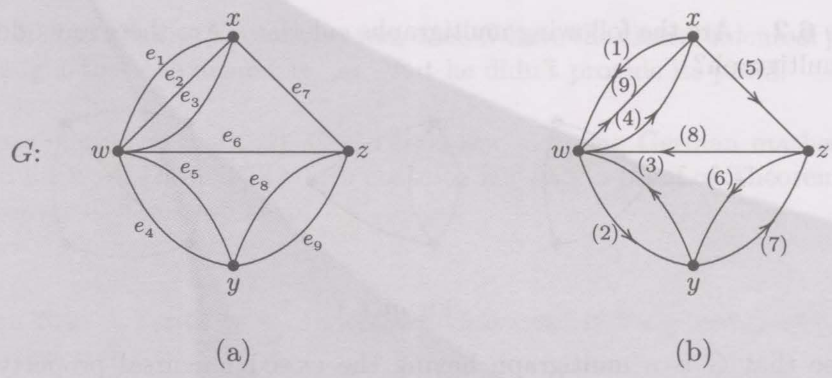


Figure 6.1

Observe that W is a closed trail (no edge is repeated) which traverses *every* edge of G. This reminds us the Königsberg Bridge Problem introduced in Section 1 [3] which asks essentially whether the multigraph of Figure 6.2 possesses a closed trail which passes through each of its edges.

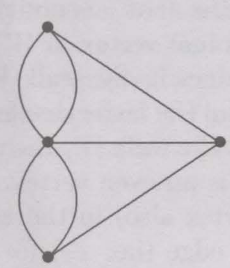


Figure 6.2

Instead of merely considering the Königsberg Bridge Problem, Euler [1] asked the following general question: what can be said about a multigraph if

($*$) it possesses a closed trail which passes through each of its edges?

In literature, the property ($*$) is often referred to as the *closed unicursal property.* In memory of Euler, such a closed trail is named a *closed Euler trail* and any multigraph which possesses a closed Euler trail is named an *eulerian multigraph.* Thus, the multigraph of Figure 6.1 (a) is an eulerian multigraph.

Exercise 6.1. Show that each of the following multigraphs is eulerian by exhibiting a closed Euler trail.

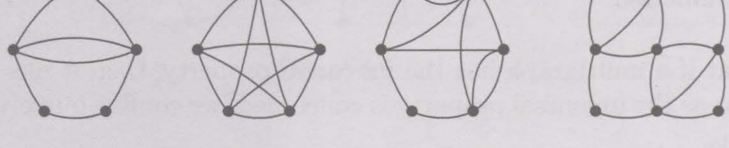


Figure 6.3

Is there any *odd* vertex (i.e., a vertex of odd degree) in each multigraph?

Graphs and Their Applications (2)

Exercise 6.2. Are the following multigraphs eulerian? Are there any odd vertices in each multigraph?

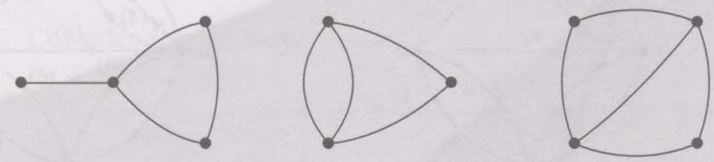


Figure 6.4

Suppose that G is a multigraph having the closed unicursal property. Then, by definition, G possesses a closed walk $W : v_1e_1v_2e_2 \ldots v_me_mv_{m+1}$, which passes through each edge of G once and exactly once. Thus $e_1, e_2, \ldots, e_m$ are the distinct edges in G, and each edge in G is one of the e_i's. Note that the vertices $v_1, v_2, \ldots, v_{m+1}$ need not be distinct (indeed, $v_1 = v_{m+1}$).

Euler now asserted that each vertex in G must be *even*. To see this, let v be an arbitrary vertex in G. Assume first that v is not the initial vertex in the closed walk W (hence v is also not the terminal vertex in W). Then each time we traverse W to visit v, there must be two edges in the walk W, say e_i and e_{i+1}, such that the former one is for us to reach v and the latter one for us to leave v. Since all the edges incident with v are contained in the walk W, the number of edges incident with v is thus even, which means that v is an even vertex. Assume now that v is the initial vertex (and so the terminal vertex also) in the walk W. That is, $v = v_1 = v_{m+1}$. For the first move, there is an edge (i.e. e_1) for us to leave v; for the last move, there is an edge (i.e., e_m) for us to return to v; and besides these, each time we visit v (if any) there must be two edges in the walk W, one for entering v and one for leaving v. Thus, again, the number of edges incident with v is even; that is, v is an even vertex.

Euler's assertion is now re-stated as follows:

Theroem 6.1. *If G is an eulerian multigraph, then each vertex in G is even.*

The negative answer to the Königsberg Bridge Problem now follows readily from Theorem 6.1. Consider the multigraph G of Figure 6.2. Since *not* every vertex in G is even (indeed, every vertex in G is odd), by Theorem 6.1, G is not eulerian.

By Theorem 6.1, it is now easy to see that all the multigraphs shown in Exercise 6.2 are not eulerian.

Note that if a multigraph has the unicursal property, then it must be connected. Thus, as far as the unicursal property is concerned, we confine ourselves to connected multigraphs.

Theorem 6.1 says that if a multigraph G has the closed unicursal property, then very vertex of G must be even. Is the converse true? That is, if G is a connected multigraph in which every vertex is even, does G have the closed unicursal property? Euler thought that the answer is 'yes', but he didn't provide its proof.

Unaware of Euler's work [1], Carl Hierholzer, a young German mathematician, published his work [2] in 1873 which contains not only a proof of Theorem 6.1, but also its converse. Thus, we have:

Theorem 6.2. *Let G be a connected multigraph. If every vertex of G is even, then G is eulerian.* □

Exercise 6.3. Prove Theorem 6.2.

Consider the multigraph G of Figure 6.5 (a) with two specified vertices u and v. Figure 6.5 (b) shows a $u - v$ trail T which passes through each edge in G, but T is not closed ($u \neq v$). In this situation, we say that G has the *open unicursal property*, T is an *open Euler trail* and G is a *semi-eulerian* multigraph. Note that u and v are the only two odd vertices in G. In general, a multigraph is said to have an *open unicursal property* or said to be *semi-eulerian* if it possesses an *open Euler trail*, i.e., an open walk which passes through each of its edges once and exactly once.

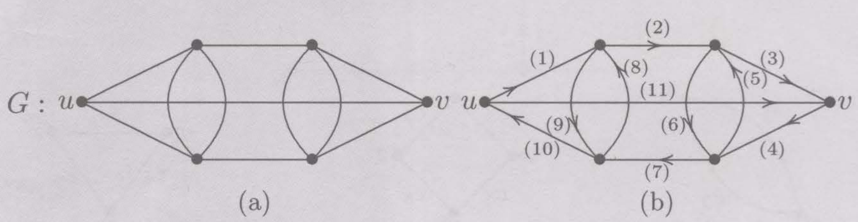


Figure 6.5

Exercise 6.4. Determine whether the following multigraphs are semi-eulerian. How many odd vertices are there in each of them?

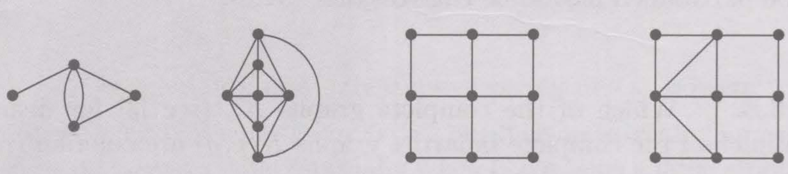


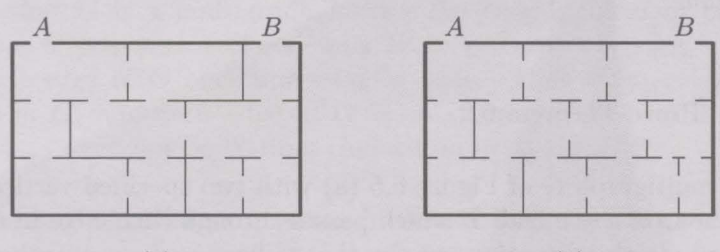
Figure 6.6

Exercise 6.5. Applying Theorems 6.1 and 6.2, or otherwise, show that a connected multigraph is semi-eulerian if and only if it contains exactly two odd vertices.

Exercise 6.6. Two halls are partitioned into small rooms for an exhibition event in two different ways as shown in (a) and (b) below, where A is the entrance and B is the exit.

(i) Is it possible for a visitor to have a route which enters at A, passes through each door once and exactly once and exits at B in partition (a)?

(ii) Explain why such a route is not available in partition (b). Which door should be closed to ensure the existence of such a route?



(a) (b)

Figure 6.7

Exercise 6.7. We have shown that the multigraph G of Figure 6.1 (a) is eulerian. Look at its edge set $E(G)$ and observe that the edges in G can be partitioned into three edge-disjoint cycles as shown below:

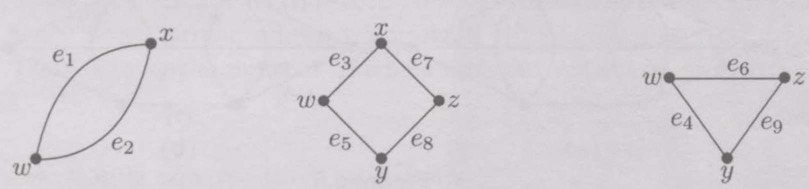


Figure 6.8

Show that, in general, a connected multigraph is eulerian if and only if all its edges can be partitioned into some edge-disjoint cycles.

Exercise 6.8. Which of the complete graphs K_n (see [3] for definition) are eulerian? Which of the complete bipartite graphs $K(p, q)$ are eulerian (resp., semi-eulerian)?

Exercise 6.9. Let G_1 and G_2 be two connected semi-eulerian multigraphs.

(i) Is it possible to form a semi-eulerian multigraph by adding a new edge joining a vertex u in G_1 and a vertex v in G_2 as shown below? If the answer is 'yes', how can this be done?

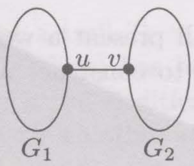


Figure 6.9

(ii) Is it possible to form an eulerian multigraph by adding two new edges, each of which joining a vertex in G_1 and a vertex in G_2? If the answer is 'yes', how can this be done?

Exercise 6.10. Let G_1 and G_2 be two connected multigraphs having $2p$ and $2q$ odd vertices respectively, where $1 \leq p \leq q$. We wish to form an eulerian multigraph from G_1 and G_2 by adding new edges, each of which joining a vertex in G_1 and a vertex in G_2. What is the *least* number of edges that should be added?

Exercise 6.11. The following graph H is not eulerian. What is the *least* number of new edges that should be added to H so that the resulting multigraph becomes eulerian? In how many ways can this be done?

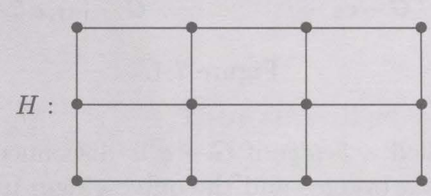


Figure 6.10

7. Fleury's Algorithm

Throughout this section, let G be a connected multigraph. Theorems 6.1 and 6.2 tell us that

$$G \text{ is eulerian if and only if every vertex in } G \text{ is even.} \tag{7.1}$$

Determine whether G is eulerian by trying our luck on searching for a closed Euler trail in G is by no means simple especially when G contains a large amount of edges. On the other hand, checking whether a vertex is even is really a small matter. Thus, the result in (7.1) enables us in reducing the amount of work to determine if G is eulerian.

Suppose we know that G is eulerian by (7.1). The next natural question is : how are we going to find a closed Euler trail in G? Unfortunately, no answer is given in (7.1). In this section, we shall present a well-known procedure, due to Fleury (before 1921), which enables us to construct a closed Euler trail in a connected eulerian multigraph efficiently.

A special type of edges and some notation will be introduced in advance. Given an edge e in G, we denote by $G - e$ the multigraph obtained by deleting e from G.

More generally, if F is a set of edges in G, we denote by $G - F$ the multigraph obtained by deleting successively the edges in F from G (see Figure 7.1).

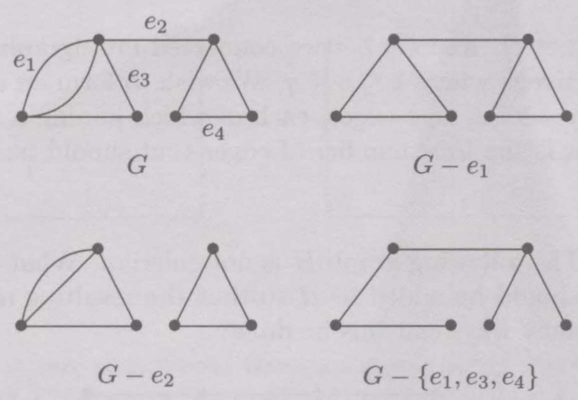


Figure 7.1

An edge e in G is called a *bridge* if $G - e$ is disconnected. Thus, as shown in Figure 7.1, the edge e_2 is a bridge (and the only bridge) in G.

Before presenting Fleury's method formally, let us mention briefly its 'idea'. We may choose any vertex, say v_0, to start off. Then select an edge f_1, say $f_1 = v_0v_1$, incident with it, and traverse from v_0 via f_1 to reach v_1. Delete f_1 from G to get $G - f_1$. Now, from v_1, select an edge f_2 in $G - f_1$, say $f_2 = v_1v_2$, incident with v_1 such that f_2 is *not* a bridge in $G - f_1$, unless there is no other alternative. Traverse from v_1 via f_2 to reach v_2. Delete f_2 from $G - f_1$ and repeat the procedure until a closed Euler trail is found. It is noted that, in the above procedure, selecting f_i from $G - \{f_i, \ldots, f_{i-1}\}$ such that f_i is *not* a bridge in $G - \{f_i, \ldots, f_{i-1}\}$, if it is available, is to ensure that no edge in G is missed from the trail formed.

We are now in a position to state the following:

Fleury's Algorithm

Given: a connected eulerian multigraph G.
Objective: to construct a closed Euler trail in G.

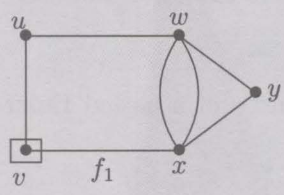
1° (Initial step) Choose an arbitrary vertex, say v_0, and set $T_0 = v_0$ (an initial trail).

2° (Inductive step) Assume that a trail $T_k = v_0 f_1 v_1 f_2 v_2 \cdots v_{k-1} f_k v_k$ (v_i's are vertices and $f_i(= v_{i-1}v_i)$'s are edges) has been constructed. Form a longer trail T_{k+1} by extending T_k with the addition of a new edge f_{k+1} such that $f_{k+1} = v_k v_{k+1}$ and, unless there is no other alternative, f_{k+1} is not a bridge in the multigraph $G - \{f_1, \ldots, f_k\}$.

3° (Ending step) Stop when step 2° cannot be implemented any further.

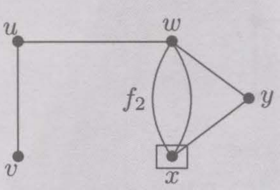
We illustrate the algorithm by the following example.

G:



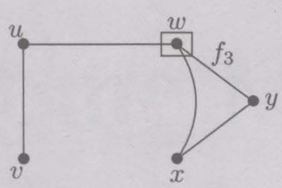
G is a given connected eulerian multigraph. We start at v and traverse from v via the edge $f_1 = vx$ to reach x.

$G - f_1$:



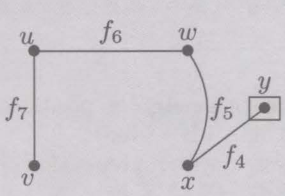
From x, choose f_2 to reach w.

$G - \{f_1, f_2\}$:



There are 3 edges incident with w. As wu is a bridge in the current multigraph, it cannot be selected. Instead, we may choose wx or wy, say $f_3 = wy$, to reach y.

$G-$ $\{f_1, f_2, f_3\}$:



From y, as there is no other choice, we follow successively f_4, f_5, f_6, f_7 and return to v.

Conclusion. The walk: $vf_1xf_2wf_3yf_4xf_5wf_6uf_7v$ is a desired closed Euler trail in G.

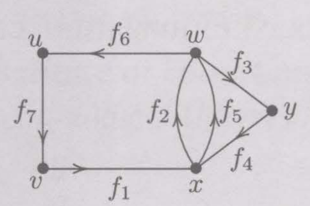


Figure 7.1

It is noted that in the third diagram shown above, should we choose wu (which is a bridge in the current graph) instead of f_3, we would return to v without passing through the three remaining edges.

We have shown how Fleury's algorithm works in constructing a closed Euler trail in a connected eulerian multigraph. Incidentally, Fleury's algorithm could be modified slightly to construct an *open* Euler trail in a connected *semi-eulerian* multigraph. In this situation, all we need to do is to choose one of the two odd vertices as the starting vertex. The algorithm itself would automatically take case of the rest and lead us to a terminal vertex which is the other odd vertex.

Exercise 7.1. Apply Fleury's algorithm to construct a closed Euler trail in the following eulerian graph.

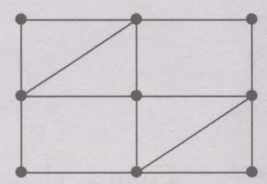


Figure 7.2

Exercise 7.2. Construct an open Euler trail in the following semi-eulerian multigraph.

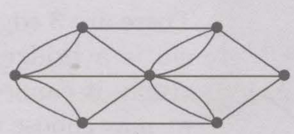


Figure 7.3

References

[1] L. Euler, The solution of a problem relating to the geometry of position, *Commentarii Academiae Scientiarum Imperialis Petropolitanae* 8(1736), 128 - 140.

[2] C. Hierholzer, On the possiblility of traversing a line-system without repetition on discontinuity, *Mathematische Annalen* 6(1873), 30-32.

[3] K. M. Koh, Graphs and their applications (1), *Mathematical Medley* 29(2) (2002), 86-94.

Graphs and Their Applications (2)