Solutions for the reattempt of the 1st test in Graphs

January 8th, 2009

Exercise 1. How many different (up to isomorphism) 3-regular simple graphs with six vertices are there?

Answer: 2. One of them is $K_{3,3}$, the other one is depicted in Fig. 1.

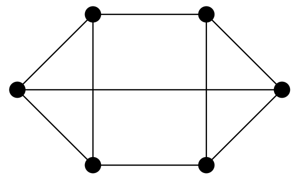

Figure 1:

There are several equally valid ways to convince oneself that there are no more possibilities. Maybe the following is the shortest argument: Two simple graphs are isomorphic iff their complement graphs are isomorphic. The complement of a three-regular graph with six vertices is a two-regular graph with six vertices. Indeed, if the number of vertices is six, then the degree of a vertex in a graph, plus the degree of this vertex in the complement graph, add up to five. Hence our task is to count the two-regular graphs with six vertices.

If the degree of each vertex in a graph is two, then this graph is a disjoint union of cycles. As each cycle must have at least three vertices, there are just two ways of distributing the six vertices in our graph: either we have one cycle (C_6) or two (two times C_3).

The complement of C_6 is depicted in Fig. 1. The complement of "two times C_3" is $K_{3,3}$.

Exercise 2. For any $n \in \mathbb{N}$ define the simple graph $G_n = (V_n, E_n)$ as follows:

- The elements of V_n are all subsets of the set $\{1, \ldots, n\}$, except the empty set.
- Two elements $A, B \in V_n$ are neighbours if and only if $A \cap B \neq \emptyset$.

For which values of n is G_n Eulerian?

Answer: only for $n = 1$. *Remark.* As G_1 consists of a single vertex and no edges, the answer "there is no such n" was also considered correct.

If $n \geq 2$ then consider the vertex $\{1\} \in V_n$. It is connected precisely to all other vertices $A \in V_n$, such that $1 \in A$. The number of such vertices A is equal to the number of subsets of the set $\{2, \ldots, n\}$, except the empty set. Hence the number of such vertices A (neighbours of $\{1\}$) is equal to $2^{n-1} - 1$, which is odd. Thus G_n is not Eulerian.

Exercise 3. Let G be a connected simple graph. Show that if the three graphs in Fig. 2 are not induced subgraphs of G, then G has a Hamiltonian path.

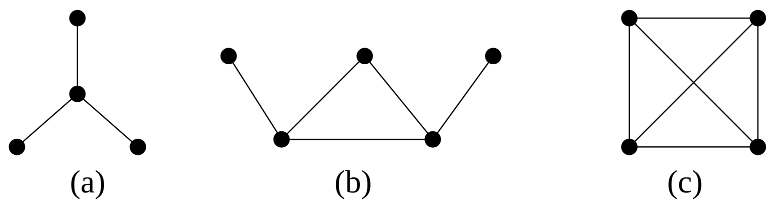

Figure 2:

Proof. Assume the opposite: there exists a connected simple graph G that has no Hamiltonian path, and also no induced subgraphs from Fig. 2. Let P be the longest path in G. By our assumption, it does not pass through all vertices. Hence there exists a vertex not on P, and by the connectivity of G, there exists a vertex v that is adjacent to some vertex c on P but is itself not on P. See Fig. 3.

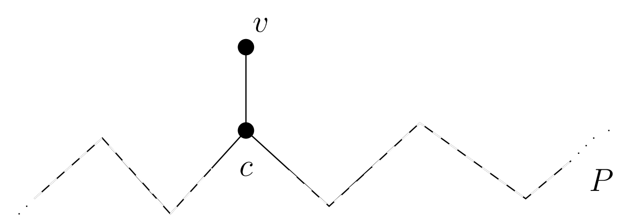

Figure 3:

If c is an end-vertex of P, then P is not the longest path in G — it can be extended by v. Hence there are vertices b and d on P, neighbouring c. See Fig. 4.

The edges (b, c), (d, c) and (v, c) form a subgraph isomorphic to the graph in Fig. 2a. As it cannot be an induced subgraph, there must be more edges with the end-vertices in $\{b, c, d, v\}$. If the edge (b, v) is present (see Fig. 5), then we can lengthen the path P by replacing the edge $b - c$ with the path

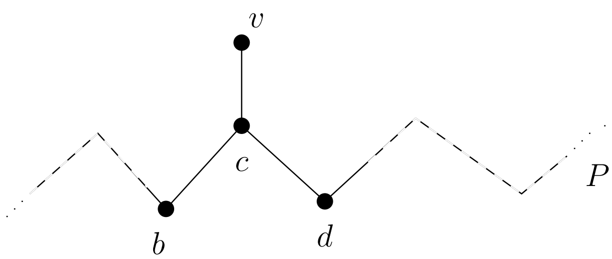

Figure 4:

$b - v - c$. The path can be similarly extended if the edge (d, c) had been present.

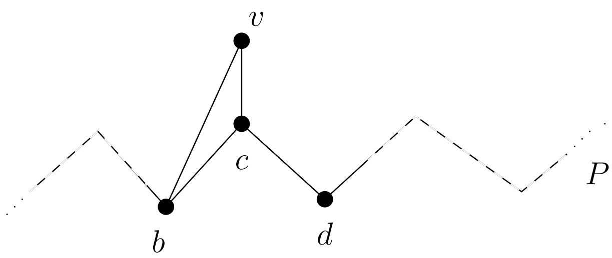

Figure 5:

Hence the edge (b, d) must be present in the graph, see Fig. 6. If b is an end-vertex of P then P can be lengthened by replacing the subpath $b-c-d$ with the path $v-c-b-d$. Similar lengthening is possible if d is an end-vertex of P. Thus there are vertices a and e on P, neighbouring respectively b and d. See Fig. 7.

The edges (a, b), (b, c), (b, d), (c, d) and (c, v) form a subgraph isomorphic to the graph in Fig. 2b. As it cannot be an induced subgraph, there must exist more edges with the end-vertices in the set $\{a, b, c, d, v\}$. We have already seen that the existence of either the edge (b, v) or the edge (d, v) would violate the maximality of the length of the path P.

If the graph contains the edge (a, v) (see Fig. 8) then the path P can be lengthened by replacing the subpath $a-b-c-d$ with the path $a-v-c-b-d$. If the graph contains the edge (a, c) (see Fig. 9) then the edges (a, c), (c, v) and (c, d) form a subgraph isomorphic to the graph in Fig. 2a. Hence there must be more edges with the end-vertices in the set $\{a, c, d, v\}$. We have already determined that the graph cannot contain the edges (a, v) or (d, v).

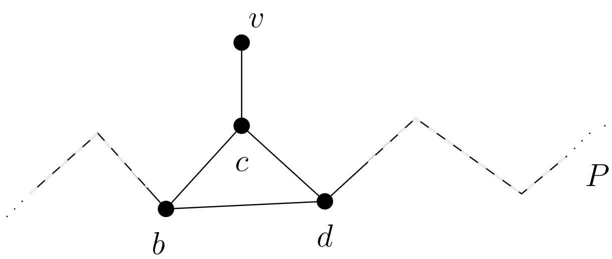

Figure 6:

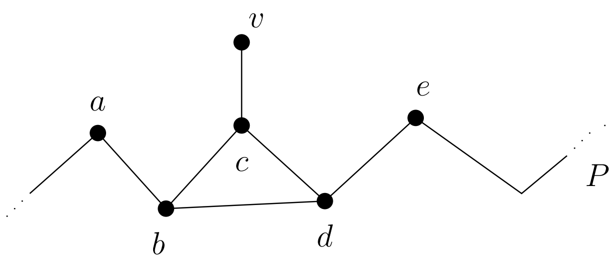

Figure 7:

But if the graph also contains the edge (a, d) then the subgraph induced by the vertices a, b, c, d is K_4, which is forbidden by Fig. 2c.

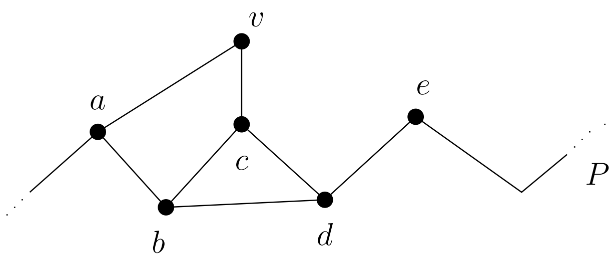

Figure 8:

Thus the graph must contain the edge (a, d) (but not the edge (a, c)). Symmetrically, if we consider the subgraph formed by edges (b, c), (b, d), (c, d), (c, v) and (c, e), we find that the graph must also contain the edge (e, b), but not contain (e, v) or (c, e), see Fig. 10. Consider now the subgraph formed by edges (a, d), (c, d) and (e, d). It is isomorphic to the graph in Fig. 2a, hence the graph must contain more edges with end-vertices in $\{a, c, d, e\}$. We have

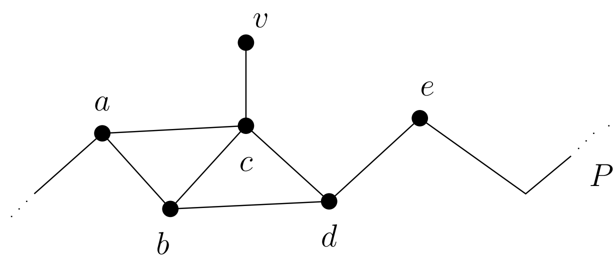

Figure 9:

already found that the edges (a, c) and (c, e) cannot be present. But if the edge (a, e) is present then the subgraph induced by the vertices a, b, d, e is isomorphic to K_4, which is forbidden by Fig. 2c.

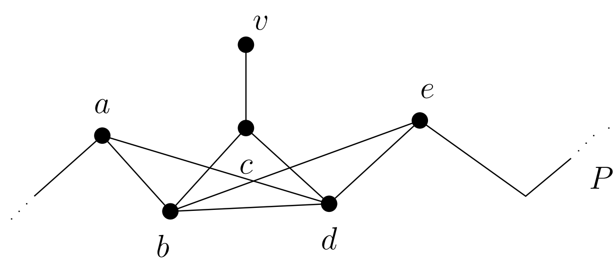

Figure 10:

Exercise 4. Consider the graph G_4 from the second exercise. For an edge e connecting the vertices A and B define its weight $w(e)$ as the sum of elements of the set $A \cap B$. Find

- a maximum-weight spanning tree;
- a minimum-weight spanning tree

of the resulting graph with edge weights.

Sketch of the solution. Just use Kruskal's algorithm.

One of the maximum-weight spanning trees is the star graph, with the universal set $\{1, 2, 3, 4\}$ in the middle and every other vertex connected directly to it. Indeed, for any $A \in V_4$, the edge connecting it to $\{1, 2, 3, 4\}$ has the maximum weight among the edges incident to A. Hence, in a run of the Kruskal algorithm, it can occur as the first considered edge that is incident to A.

It is actually possible to also describe *all* maximum-weight spanning trees of G_n. They are those, where each vertex (except $\{1, \ldots, n\}$) is connected to exactly one of its supersets (and to any number of its subsets).

For the minimum-weight spanning tree, again imagine how Kruskal's algorithm works. First we get all edges of weight one, then all edges of weight two, etc. Hence, the following is a possible minimum-weight spanning tree of G_n:

- Each vertex A, where $1 \in A$, is connected to $\{1\}$.
- The vertex $\{2\}$ is connected to $\{1, 2\}$.
- Each vertex A, where $2 \in A$ and $1 \notin A$, is connected to $\{2\}$.
- The vertex $\{3\}$ is connected to $\{1, 3\}$.
- Each vertex A, where $3 \in A$, and $1, 2 \notin A$, is connected to $\{3\}$.
- The vertex $\{4\}$ is connected to $\{1, 4\}$.
- etc. (actually, for the graph G_4, we stop here)