MTH 325 – Exam 2 Solutions

**Skill 1: (CORE) I can outline a proof by mathematical induction.**

Consider the following proposition: For every integer , is divisible by 11. (Note, this is 10 raised to the 2*n*-1 power, plus 1. The exponent on 10 is 2*n*-1; the additional +1 is not in the exponent.)

1. State the value of *n* that corresponds to the base case, then prove that the base case holds.

The base case is n = 1. In this case, . This is clearly divisible by 11 so the base case holds.

1. Clearly state the inductive hypothesis. Your answer should be phrased as a complete sentence. (No explanation is required here; simply state the inductive hypothesis.)

Assume that for some positive integer *k*, is divisible by 11.

1. Clearly state what you would need to prove, after assuming the inductive hypothesis. Your answer should be phrased as a complete sentence. (You do not need to give a completed proof the statement; simply state what you would need to prove.)

Prove that is divisible by 11. Note, if you expand the algebra in the exponent this is the same as proving that is divisible by 11.

**Skill 2: (CORE) I can outline a proof using direct, contrapositive, and indirect approaches.**

Consider the following proposition: Suppose that *G* is a graph. **If the degree of each vertex of *G* is even, then *G* has an Euler circuit.**

1. Clearly state what you would assume and what you would need to prove, if you were to prove this statement with a *direct proof*. (No further explanation is necessary.)
2. Clearly state what you would assume and what you would need to prove, if you were to prove this statement with a *proof by contrapositive*. (No further explanation is necessary.)
3. Clearly state all assumptions you would make, if you were to prove this statement with a *proof by contradiction* (also known as an *indirect proof*). (No further explanation is necessary.)

Results are summarized below:

|  |  |  |
| --- | --- | --- |
| Proof method | Assume… | Prove… |
| Direct | The degree of each vertex of G is even | G has an Euler circuit |
| Contrapositive | G does not have an Euler circuit | ***There exists*** a vertex in G whose degree is odd |
| Indirect | The degree of each vertex of G is even but G does not have an Euler circuit | (n/a) |

Note in the contrapositive proof, the negation of “The degree of each vertex of G is even” is not “The degree of each vertex in G is odd”. Both of those statements might be false at the same time, so they are not negations of each other.

**Skill 3 (CORE) I can represent a graph in different ways, determine information (degree, degree sequence, paths of given length, etc.) about a graph using different representations, and give examples of graphs with specified properties.**

Consider the graph *G* given by this adjacency matrix:

A number grid with numbers

Description automatically generated with medium confidence

1. Assume that the vertices are labeled *a, b, c, d, e, f* and the rows and columns of this matrix correspond to that ordering (so the first row and column represent vertex *a*, the second represent vertex *b* and so on). State the Python dictionary for this graph.

For simplicity I will leave off the quote marks that Python normally wants to have around strings.

**{a: [b,e], b: [a,f], c: [d], d: [c,f], e: [a,f], f: [b,d,e]}.**

1. State the degree of each vertex. You don’t need to explain your answers here, just make sure they are right.

**Deg(a) = 2, deg(b) = 2, deg(c) = 1, deg(d) = 2, deg( e) = 2, deg(f) = 3.**

1. Give the number of edges in the graph and show your work or explain your reasoning.

The degree sum is 2 + 2 + 1 + 2 + 2 + 3 = 12. The Handshake Lemma says this is twice the number of edges, **so the number of edges is 6.**

1. Give an example of a cycle of length 4 in this graph. If no such cycle exists, say so and explain your reasoning.

One such cycle is: **a, b, f, e, a.**

**Skill 4: I can determine whether a graph has an Euler trail or Euler circuit, and whether a graph has a Hamiltonian path or circuit.**

Consider the graph *G* shown below:



1. Determine whether this graph has an Euler trail, and explain how you know.

To have an Euler trail, there must be at most two vertices of odd degree. But that is not the case here since 1, 2, 3, and 4 all have odd degree. **So there’s no Euler trail**.

1. Determine whether this graph has an Euler circuit, and explain how you know.

To have an Euler circuit, *all* of the vertices must have even degree, and above we showed that wasn’t the case. **So there’s no Euler circuit.**

1. Determine whether this graph has a Hamilton path, and explain how you know.

There is at least one Hamilton path: **1, 7, 6, 5, 3, 4, 8, 2.**

1. Determine whether this graph has a Hamilton circuit, and explain how you know.

There is at least one Hamilton circuit: **1, 7, 6, 5, 3, 4, 8, 2, 1.**

**Skill 5: I can use a greedy algorithm to find a vertex coloring for a graph, and I can determine a graph's chromatic number.**

Consider the graph below:



1. Use a greedy algorithm to construct a proper vertex coloring for this graph. For the initial ordering of the vertices, use the degree of the vertices from highest degree to lowest, and use numerical ordering in the case of a tie. Your work should consist of a list of vertices in the order in which they are considered; and the color assigned to each one, given as a non-negative integer.

The vertices to visit, in order are: **7, 4, 6, 2, 3, 8, 5, 9, 1, 10**.

* Assign color 0 to vertex 7.
* Assign color 1 to vertex 4 since it’s adjacent to 7.
* Assign color 0 to vertex 6 – this is the lowest valued color not already assigned to a neighbor of vertex 0.
* Assign color 1 to vertex 2 – this is the lowest valued color not already assigned to a neighbor of vertex 2.
* Assign color 2 to vertex 3 – this is the lowest valued color not already assigned to a neighbor of vertex 3.
* Assign color 2 to vertex 8 – this is the lowest valued color not already assigned to a neighbor of vertex 8.
* Assign color 1 to vertex 5 – this is the lowest valued color not already assigned to a neighbor of vertex 5.
* Assign color 2 to vertex 9 – this is the lowest valued color not already assigned to a neighbor of vertex 9.
* Assign color 1 to vertex 1 – this is the lowest valued color not already assigned to a neighbor of vertex 1.
* Assign color 1 to vertex 10 – this is the lowest valued color not already assigned to a neighbor of vertex 10.

The above info is an adequate response. As a Python dictionary like networkX would produce, the coloring would be **{1: 1, 2: 1, 3: 2, 4: 1, 5: 1, 6: 0, 7: 0, 8: 2, 9: 2, 10: 1}.**

1. State the chromatic number of this graph and explain your reasoning.

**The chromatic number is 3**. The above coloring uses 3 colors, and fewer colors are not possible because of the triangles (copies of ) present in the graph.

**Skill 6: I can determine whether two graphs are isomorphic; I can give an explicit isomorphism if they are, and an explanation if they are not.**

Given the two graphs below, state whether they are isomorphic. If they are isomorphic, give an explicit isomorphism between the two and explain why your mapping is really an isomorphism. If they are not isomorphic, give an explicit isomorphism invariant property that one has but the other does not have.

A diagram of a triangle with lines and dots

Description automatically generated with medium confidenceThese two graphs **are not isomorphic**. There are multiple ways to justify this: Graph G has a cycle of length 4 and H does not; graph H has a vertex of degree 4 and G does not; graph H has a vertex of degree 1 and G does not; graph G has two vertices of degree 3 but H has only one; and so on and so forth.

**Skill 7 (CORE): I can determine whether a graph is a tree and state information about it.**

1. Draw (or give a Python dictionary for) a graph that has degree sequence 2, 2, 2, 1, 1 that is *not* a tree. If no such example is possible, explain why.

**An example of such a graph is a copy of together with a single disconnected edge**, which is not a tree because (1) it contains a cycle and (2) it’s not connected. (This example was discussed in class.)

1. How many vertices does a tree with 100 edges have? State your answer; if there is not enough information to arrive at an answer, say so. No explanation needed either way.

In any tree, the number of edges is one less than the number of vertices. So if there are 100 edges, there must be **101 vertices.**

1. Consider the rooted tree shown below, with vertex 6 as the root. State the following. (No explanation needed but be sure to label your answers.)
   1. The children of vertex 9 **– 7 and 5**
   2. The parent(s) of vertex 3 **– vertex 8**
   3. The height of the tree – **This is the length of the longest path from the root to a leaf, so 3.**
   4. The leaves of the tree – **2, 3, 1, 7, 10, and 4**



**Skill 8: I can use Prim's Algorithm and Kruskal's Algorithm to construct a minimum spanning tree for a weighted graph.**

Consider the weighted graph below:

A black background with white dots

Description automatically generated

1. Use Prim’s algorithm, with initial vertex 3, to construct a minimum spanning tree for this graph. Your work should consist of a list of edges in the tree, given in the order in which they are added.

Starting with vertex 3, we grow the tree by adding a minimum-weight edge at each step that connects one vertex in the tree to another vertex outside the tree. The table below shows the tree at the beginning of each step and which edge is added at that step.

|  |  |  |
| --- | --- | --- |
| **Step** | **Current MST** | **Edge to add** |
| 1 | [ ] (empty) | **{0,3}** |
| 2 | [{0,3}] | **{0,4}** |
| 3 | [{0,3}, {0,4}] | **{4,7}** |
| 4 | [{0,3}, {0,4}, {4,7}] | **{4,2}** |
| 5 | [{0,3}, {0,4}, {4,7}, {4,2}] | **{2,5}** |
| 6 | [{0,3}, {0,4}, {4,7}, {4,2}, {2,5}] | **{6,7}** |
| 7 | [{0,3}, {0,4}, {4,7}, {4,2}, {2,5}, {6,7}] | **{0,6}** |

This results in the tree [{0,3}, {0,4}, {4,7}, {4,2}, {2,5}, {6,7}, {0,6}] which has total weight 23. It’s not necessary to draw the MST, but it looks like this:

A diagram of a circle with circles and lines

Description automatically generated

1. Use Kruskal’s algorithm to construct a minimum spanning tree for this graph. Your work should consist of a list of edges in the tree, given in the order in which they are added.

The order in which the edges are added is: {4,7}, {2,5}, {2,4} (or {0,3}), {0,3} (or {2,4}), {0,4} (or {6,7}), {6,7} (or {0,4}), {0,1}. This happens to give the same MST as Prim’s Algorithm this time.