

# Solutions to Chapter 10, Susanna Epp Discrete Math 5th Edition

<https://github.com/spamegg1>

November 16, 2023

## Contents

<b>1</b>	<b>Exercise Set 10.1</b>	<b>10</b>
1.1	Exercise 1	10
1.1.1	(a)	10
1.1.2	(b)	10
1.1.3	(c)	10
1.1.4	(d)	10
1.1.5	(e)	10
1.1.6	(f)	11
1.2	Exercise 2	11
1.2.1	(a)	11
1.2.2	(b)	11
1.2.3	(c)	11
1.2.4	(d)	11
1.2.5	(e)	11
1.2.6	(f)	12
1.3	Exercise 3	12
1.3.1	(a)	12
1.3.2	(b)	12
1.4	Exercise 4	12
1.4.1	(a)	12
1.4.2	(b)	12
1.4.3	(c)	13
1.5	Exercise 5	13
1.5.1	(a)	13
1.5.2	(b)	13
1.5.3	(c)	13
1.6	Exercise 6	13
1.6.1	(a)	13
1.6.2	(b)	14
1.6.3	(c)	14

1.7	Exercise 7	14
1.8	Exercise 8	15
1.8.1	(a)	15
1.8.2	(b)	15
1.8.3	(c)	15
1.8.4	(d)	16
1.9	Exercise 9	16
1.9.1	(a)	16
1.9.2	(b)	16
1.9.3	(c)	16
1.10	Exercise 10	16
1.11	Exercise 11	16
1.12	Exercise 12	17
1.13	Exercise 13	17
1.14	Exercise 14	17
1.15	Exercise 15	18
1.16	Exercise 16	18
1.17	Exercise 17	18
1.18	Exercise 18	18
1.19	Exercise 19	19
1.20	Exercise 20	19
1.21	Exercise 21	19
1.22	Exercise 22	20
1.23	Exercise 23	20
1.23.1	(a)	20
1.23.2	(b)	21
1.23.3	(c)	21
1.24	Exercise 24	22
1.24.1	(a)	22
1.24.2	(b)	22
1.25	Exercise 25	22
1.25.1	(a)	22
1.25.2	(b)	22
1.26	Exercise 26	23
1.26.1	(a)	23
1.26.2	(b)	23
1.27	Exercise 27	23
1.28	Exercise 28	24
1.29	Exercise 29	24
1.30	Exercise 30	24
1.31	Exercise 31	25
1.32	Exercise 32	25
1.33	Exercise 33	25
1.34	Exercise 34	26
1.35	Exercise 35	26
1.36	Exercise 36	26

1.37	Exercise 37	27
1.38	Exercise 38	27
1.39	Exercise 39	27
1.40	Exercise 40	28
1.41	Exercise 41	28
1.42	Exercise 42	28
1.43	Exercise 43	29
1.43.1	(a)	29
1.43.2	(b)	30
1.44	Exercise 44	30
1.45	Exercise 45	30
1.46	Exercise 46	31
1.47	Exercise 47	31
1.48	Exercise 48	31
1.49	Exercise 49	31
1.50	Exercise 50	32
1.51	Exercise 51	32
1.52	Exercise 52	32
1.53	Exercise 53	33
1.54	Exercise 54	33
1.55	Exercise 55	34
1.56	Exercise 56	34
1.56.1	(a)	34
1.56.2	(b)	34
1.57	Exercise 57	35
<b>2</b>	<b>Exercise Set 10.2</b>	<b>36</b>
2.1	Exercise 1	36
2.1.1	()	36
2.2	Exercise 2	36
2.2.1	()	36
2.3	Exercise 3	36
2.3.1	()	36
2.4	Exercise 4	36
2.4.1	()	36
2.5	Exercise 5	36
2.5.1	()	36
2.6	Exercise 6	36
2.6.1	()	36
2.7	Exercise 7	36
2.7.1	()	36
2.8	Exercise 8	37
2.8.1	()	37
2.9	Exercise 9	37
2.9.1	()	37
2.10	Exercise 10	37

2.10.1	()	37
2.11	Exercise 11	37
2.11.1	()	37
2.12	Exercise 12	37
2.12.1	()	37
2.13	Exercise 13	37
2.13.1	()	37
2.14	Exercise 14	37
2.14.1	()	37
2.15	Exercise 15	37
2.15.1	()	37
2.16	Exercise 16	38
2.16.1	()	38
2.17	Exercise 17	38
2.17.1	()	38
2.18	Exercise 18	38
2.18.1	()	38
2.19	Exercise 19	38
2.19.1	()	38
2.20	Exercise 20	38
2.20.1	()	38
2.21	Exercise 21	38
2.21.1	()	38
2.22	Exercise 22	38
2.22.1	()	38
2.23	Exercise 23	38
2.23.1	()	38
<b>3</b>	<b>Exercise Set 10.3</b>	<b>39</b>
3.1	Exercise 1	39
3.1.1	()	39
3.2	Exercise 2	39
3.2.1	()	39
3.3	Exercise 3	39
3.3.1	()	39
3.4	Exercise 4	39
3.4.1	()	39
3.5	Exercise 5	39
3.5.1	()	39
3.6	Exercise 6	39
3.6.1	()	39
3.7	Exercise 7	39
3.7.1	()	39
3.8	Exercise 8	39
3.8.1	()	39
3.9	Exercise 9	40

3.9.1	()	40
3.10	Exercise 10	40
3.10.1	()	40
3.11	Exercise 11	40
3.11.1	()	40
3.12	Exercise 12	40
3.12.1	()	40
3.13	Exercise 13	40
3.13.1	()	40
3.14	Exercise 14	40
3.14.1	()	40
3.15	Exercise 15	40
3.15.1	()	40
3.16	Exercise 16	40
3.16.1	()	40
3.17	Exercise 17	41
3.17.1	()	41
3.18	Exercise 18	41
3.18.1	()	41
3.19	Exercise 19	41
3.19.1	()	41
3.20	Exercise 20	41
3.20.1	()	41
3.21	Exercise 21	41
3.21.1	()	41
3.22	Exercise 22	41
3.22.1	()	41
3.23	Exercise 23	41
3.23.1	()	41
3.24	Exercise 24	41
3.24.1	()	41
3.25	Exercise 25	42
3.25.1	()	42
3.26	Exercise 26	42
3.26.1	()	42
3.27	Exercise 27	42
3.27.1	()	42
3.28	Exercise 28	42
3.28.1	()	42
3.29	Exercise 29	42
3.29.1	()	42
3.30	Exercise 30	42
3.30.1	()	42
<b>4</b>	<b>Exercise Set 10.4</b>	<b>42</b>
4.1	Exercise 1	42

4.1.1	()	42
4.2	Exercise 2	42
4.2.1	()	42
4.3	Exercise 3	43
4.3.1	()	43
4.4	Exercise 4	43
4.4.1	()	43
4.5	Exercise 5	43
4.5.1	()	43
4.6	Exercise 6	43
4.6.1	()	43
4.7	Exercise 7	43
4.7.1	()	43
4.8	Exercise 8	43
4.8.1	()	43
4.9	Exercise 9	43
4.9.1	()	43
4.10	Exercise 10	43
4.10.1	()	43
4.11	Exercise 11	44
4.11.1	()	44
4.12	Exercise 12	44
4.12.1	()	44
4.13	Exercise 13	44
4.13.1	()	44
4.14	Exercise 14	44
4.14.1	()	44
4.15	Exercise 15	44
4.15.1	()	44
4.16	Exercise 16	44
4.16.1	()	44
4.17	Exercise 17	44
4.17.1	()	44
4.18	Exercise 18	44
4.18.1	()	44
4.19	Exercise 19	45
4.19.1	()	45
4.20	Exercise 20	45
4.20.1	()	45
4.21	Exercise 21	45
4.21.1	()	45
4.22	Exercise 22	45
4.22.1	()	45
4.23	Exercise 23	45
4.23.1	()	45
4.24	Exercise 24	45

4.24.1	()	45
4.25	Exercise 25	45
4.25.1	()	45
4.26	Exercise 26	45
4.26.1	()	45
4.27	Exercise 27	46
4.27.1	()	46
4.28	Exercise 28	46
4.28.1	()	46
4.29	Exercise 29	46
4.29.1	()	46
4.30	Exercise 30	46
4.30.1	()	46
4.31	Exercise 31	46
4.31.1	()	46
<b>5</b>	<b>Exercise Set 10.5</b>	<b>46</b>
5.1	Exercise 1	46
5.1.1	()	46
5.2	Exercise 2	46
5.2.1	()	46
5.3	Exercise 3	46
5.3.1	()	46
5.4	Exercise 4	47
5.4.1	()	47
5.5	Exercise 5	47
5.5.1	()	47
5.6	Exercise 6	47
5.6.1	()	47
5.7	Exercise 7	47
5.7.1	()	47
5.8	Exercise 8	47
5.8.1	()	47
5.9	Exercise 9	47
5.9.1	()	47
5.10	Exercise 10	47
5.10.1	()	47
5.11	Exercise 11	47
5.11.1	()	47
5.12	Exercise 12	48
5.12.1	()	48
5.13	Exercise 13	48
5.13.1	()	48
5.14	Exercise 14	48
5.14.1	()	48
5.15	Exercise 15	48

5.15.1	()	48
5.16	Exercise 16	48
5.16.1	()	48
5.17	Exercise 17	48
5.17.1	()	48
5.18	Exercise 18	48
5.18.1	()	48
5.19	Exercise 19	48
5.19.1	()	48
5.20	Exercise 20	49
5.20.1	()	49
5.21	Exercise 21	49
5.21.1	()	49
5.22	Exercise 22	49
5.22.1	()	49
5.23	Exercise 23	49
5.23.1	()	49
5.24	Exercise 24	49
5.24.1	()	49
5.25	Exercise 25	49
5.25.1	()	49
<b>6</b>	<b>Exercise Set 10.6</b>	<b>49</b>
6.1	Exercise 1	49
6.1.1	()	49
6.2	Exercise 2	49
6.2.1	()	49
6.3	Exercise 3	50
6.3.1	()	50
6.4	Exercise 4	50
6.4.1	()	50
6.5	Exercise 5	50
6.5.1	()	50
6.6	Exercise 6	50
6.6.1	()	50
6.7	Exercise 7	50
6.7.1	()	50
6.8	Exercise 8	50
6.8.1	()	50
6.9	Exercise 9	50
6.9.1	()	50
6.10	Exercise 10	50
6.10.1	()	50
6.11	Exercise 11	51
6.11.1	()	51
6.12	Exercise 12	51

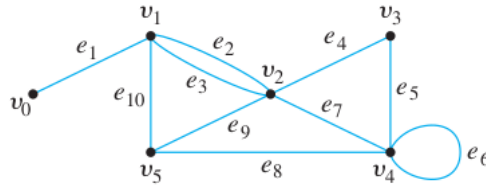


6.12.1	()	51
6.13	Exercise 13	51
6.13.1	()	51
6.14	Exercise 14	51
6.14.1	()	51
6.15	Exercise 15	51
6.15.1	()	51
6.16	Exercise 16	51
6.16.1	()	51
6.17	Exercise 17	51
6.17.1	()	51
6.18	Exercise 18	51
6.18.1	()	51
6.19	Exercise 19	52
6.19.1	()	52
6.20	Exercise 20	52
6.20.1	()	52
6.21	Exercise 21	52
6.21.1	()	52
6.22	Exercise 22	52
6.22.1	()	52
6.23	Exercise 23	52
6.23.1	()	52
6.24	Exercise 24	52
6.24.1	()	52
6.25	Exercise 25	52
6.25.1	()	52
6.26	Exercise 26	52
6.26.1	()	52
6.27	Exercise 27	53
6.27.1	()	53
6.28	Exercise 28	53
6.28.1	()	53
6.29	Exercise 29	53
6.29.1	()	53
6.30	Exercise 30	53
6.30.1	()	53
6.31	Exercise 31	53
6.31.1	()	53

# 1 Exercise Set 10.1

## 1.1 Exercise 1

In the graph below, determine whether the following walks are trails, paths, closed walks, circuits, simple circuits, or just walks.



### 1.1.1 (a)

$v_0 e_1 v_1 e_{10} v_5 e_9 v_2 e_2 v_1$

*Proof.* trail (no repeated edge), not a path (has a repeated vertex,  $v_1$ ), not a circuit

□

### 1.1.2 (b)

$v_4 e_7 v_2 e_9 v_5 e_{10} v_1 e_3 v_2 e_9 v_5$

*Proof.* walk, not a trail (has a repeated edge,  $e_9$ ), not a circuit

□

### 1.1.3 (c)

$v_2$

*Proof.* closed walk (starts and ends at the same vertex), trail (no repeated edge since no edge), not a path or a circuit (since no edge)

□

### 1.1.4 (d)

$v_5 v_2 v_3 v_4 v_4 v_5$

*Proof.* circuit, not a simple circuit (repeated vertex,  $v_4$ )

□

### 1.1.5 (e)

$v_2 v_3 v_4 v_5 v_2 v_4 v_3 v_2$

*Proof.* closed walk (starts and ends at the same vertex but has repeated edges,  $\{v_2, v_3\}$  and  $\{v_3, v_4\}$ )

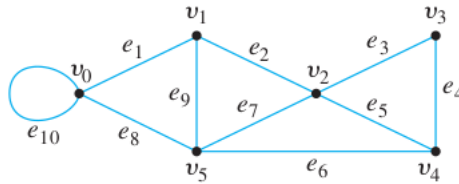
□

### 1.1.6 (f)

$e_5 e_8 e_{10} e_3$

*Proof.* path □

## 1.2 Exercise 2



In the graph, determine whether the following walks are trails, paths, closed walks, circuits, simple circuits, or just walks.

### 1.2.1 (a)

$v_1 e_2 v_2 e_3 v_3 e_4 v_4 e_5 v_2 e_2 v_1 e_1 v_0$

*Proof.* walk (not closed), not a trail or circuit (has repeated edge  $e_2$ ), not a path (has repeated vertex  $v_2$ ), □

### 1.2.2 (b)

$v_2 v_3 v_4 v_5 v_2$

*Proof.* simple circuit (only has the first + last vertex repeated, no repeated edge) □

### 1.2.3 (c)

$v_4 v_2 v_3 v_4 v_5 v_2 v_4$

*Proof.* closed walk, not a trail, path, circuit (has repeated edge  $e_5$  and vertex  $v_2$ ) □

### 1.2.4 (d)

$v_2 v_1 v_5 v_2 v_3 v_4 v_2$

*Proof.* circuit, not simple circuit (has non-first, non-last vertex repeated:  $v_2$ ) □

### 1.2.5 (e)

$v_0 v_5 v_2 v_3 v_4 v_2 v_1$

*Proof.* trail (no repeated edge), not a path (repeated vertex  $v_2$ ), not a closed walk or circuit □

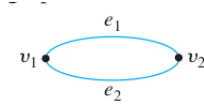
### 1.2.6 (f)

$v_5v_4v_2v_1$

*Proof.* path (no repeated vertex), not a closed walk or circuit □

## 1.3 Exercise 3

Let  $G$  be the graph



and consider the walk  $v_1e_1v_2e_2v_1$ .

### 1.3.1 (a)

Can this walk be written unambiguously as  $v_1v_2v_1$ ? Why?

*Proof.* No, because  $v_1v_2v_1$  could also refer to  $v_1e_2v_2e_1v_1$ . □

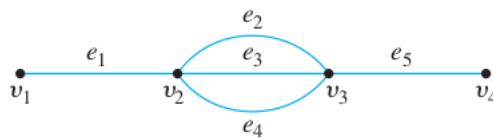
### 1.3.2 (b)

Can this walk be written unambiguously as  $e_1e_2$ ? Why?

*Proof.* Yes. □

## 1.4 Exercise 4

Consider the following graph:



### 1.4.1 (a)

How many paths are there from  $v_1$  to  $v_4$ ?

*Proof.* 3:  $e_1e_2e_5, e_1e_3e_5, e_1e_4e_5$ . □

### 1.4.2 (b)

How many trails are there from  $v_1$  to  $v_4$ ?

*Proof.*  $3! + 3 = 9$  (The three paths from part (a) are also trails, and there are an additional  $3!$  trails with vertices  $v_1, v_2, v_3, v_2, v_3, v_4$ . The reason is that from  $v_2$  there are 3 choices of an edge to go to  $v_3$ , then 2 choices of a different edge to go back to  $v_2$ , and then 1 choice of a different edge to return to  $v_3$ . □

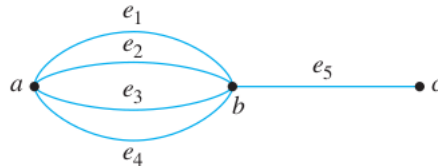
### 1.4.3 (c)

How many walks are there from  $v_1$  to  $v_4$ ?

*Proof.* Infinitely many. (Since a walk may have repeated edges, a walk from  $v_1$  to  $v_4$  may contain an arbitrarily large number of repetitions of edges joining a pair of vertices along the way.)  $\square$

## 1.5 Exercise 5

Consider the following graph:



### 1.5.1 (a)

How many paths are there from  $a$  to  $c$ ?

*Proof.* 4:  $e_1e_5, e_2e_5, e_3e_5, e_4e_5$ .  $\square$

### 1.5.2 (b)

How many trails are there from  $a$  to  $c$ ?

*Proof.*  $4! + 4 = 28$  (The 4 paths from part (a) are also trails, and there are an additional  $4!$  trails with vertices  $a, b, a, b, c$ . The reason is that from  $a$  there are 4 choices of an edge to go to  $b$ , then 3 choices of a different edge to go back to  $a$ , then 2 choices of a different edge to go back to  $b$ , and then 1 choice of a different edge to go to  $c$ .  $\square$

### 1.5.3 (c)

How many walks are there from  $a$  to  $c$ ?

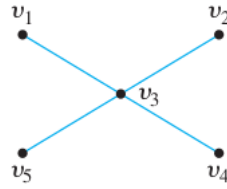
*Proof.* Infinitely many. (Since a walk may have repeated edges, a walk from  $a$  to  $c$  may contain an arbitrarily large number of repetitions of edges joining a pair of vertices along the way.)  $\square$

## 1.6 Exercise 6

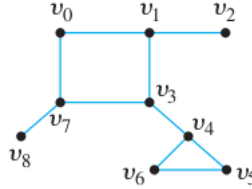
An edge whose removal disconnects the graph of which it is a part is called a bridge. Find all bridges for each of the graphs at the top of the next page.

### 1.6.1 (a)

*Proof.*  $\{v_1, v_3\}, \{v_2, v_3\}, \{v_4, v_3\}$ , and  $\{v_5, v_3\}$  are all the bridges.  $\square$

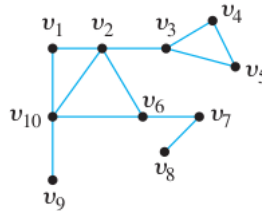


### 1.6.2 (b)



*Proof.*  $\{v_1, v_2\}, \{v_3, v_4\}, \{v_7, v_8\}$  are all the bridges. □

### 1.6.3 (c)



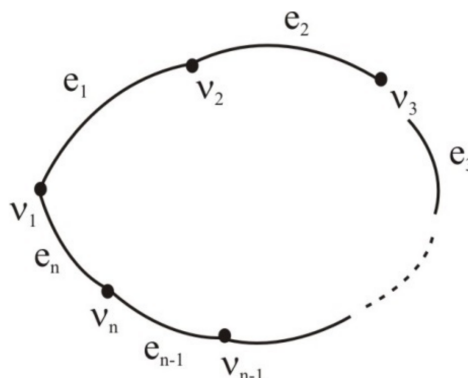
*Proof.*  $\{v_2, v_3\}, \{v_6, v_7\}, \{v_7, v_8\}, \{v_9, v_{10}\}$  are all the bridges. □

## 1.7 Exercise 7

Given any positive integer  $n$ , (a) find a connected graph with  $n$  edges such that removal of just one edge disconnects the graph; (b) find a connected graph with  $n$  edges that cannot be disconnected by the removal of any single edge.

*Proof.* (a) Consider a line graph with  $n$  edges and  $n + 1$  vertices that looks like:  $\cdot - \cdot - \dots - \cdot - \cdot : v_1 e_1 v_2 e_2 \dots v_n e_n v_{n+1}$ . Then removing any one edge will break the line and disconnect the graph.

(b) Consider a closed loop graph with  $n$  edges and  $n$  vertices:  $v_1 e_1 v_2 e_2 \dots v_n e_n v_1$

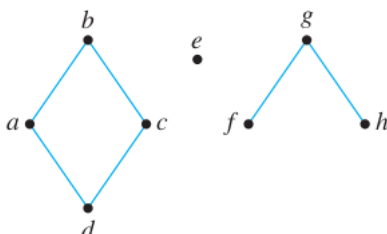


(so the last edge  $e_n$  connects back to the first vertex  $v_1$ , closing the loop). Then removing any one edge will break the loop, but it will remain connected.  $\square$

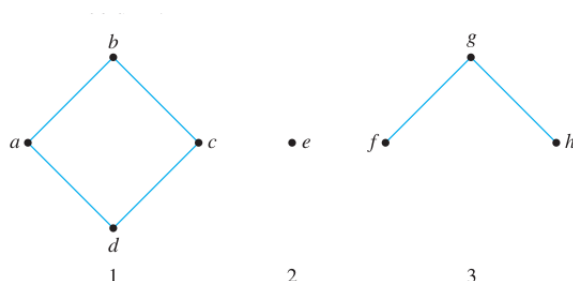
## 1.8 Exercise 8

Find the number of connected components for each of the following graphs.

### 1.8.1 (a)

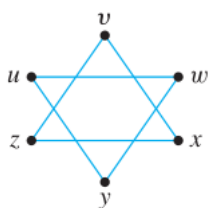


*Proof.* Three connected components, as shown in the next column.



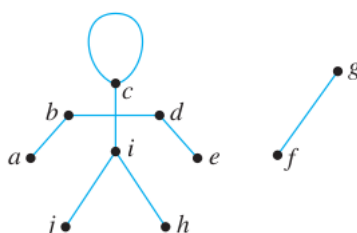
$\square$

### 1.8.2 (b)



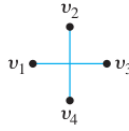
*Proof.* Two connected components: the two “triangles”  $v - x - z$  and  $u - w - y$ .  $\square$

### 1.8.3 (c)



*Proof.* Three connected components: the “arms and shoulders” part  $a - b - d - e$ , the “head-torso-legs” part  $c - i - j - h$  and the part  $f - g$ .  $\square$

### 1.8.4 (d)



*Proof.* Two connected components:  $v_1 - v_3$  and  $v_2 - v_4$ . □

## 1.9 Exercise 9

Each of (a)–(c) describes a graph. In each case answer yes, no, or not necessarily to this question: Does the graph have an Euler circuit? Justify your answers.

### 1.9.1 (a)

$G$  is a connected graph with five vertices of degrees 2, 2, 3, 3, and 4.

*Proof.* No. This graph has two vertices of odd degree, whereas all vertices of a graph with an Euler circuit have even degree. □

### 1.9.2 (b)

$G$  is a connected graph with five vertices of degrees 2, 2, 4, 4, and 6.

*Proof.* Yes (connected, all vertices have even degree, so by Theorem 10.1.4). □

### 1.9.3 (c)

$G$  is a graph with five vertices of degrees 2, 2, 4, 4, and 6.

*Proof.* Not necessarily (if  $G$  is connected then yes by Theorem 10.1.4). □

## 1.10 Exercise 10

The solution for Example 10.1.6 shows a graph for which every vertex has even degree but which does not have an Euler circuit. Give another example of a graph satisfying these conditions.

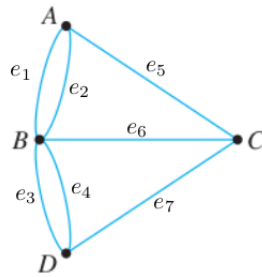
*Proof.* Just take two disconnected squares: graphs with 4 vertices each. Considered together as one graph, each vertex has degree 2 but there is no Euler circuit. □

## 1.11 Exercise 11

Is it possible for a citizen of Königsberg to make a tour of the city and cross each bridge exactly twice? (See figure below.) Explain.

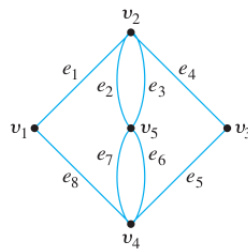
*Proof.*  $Ae_1Be_2Ae_5Ce_6Be_3De_4Be_6Ce_7De_3Be_4De_7Ce_5Ae_1Be_2A$ . (See below.) □





Determine which of the graphs in 12–17 have Euler circuits. If the graph does not have an Euler circuit, explain why not. If it does have an Euler circuit, describe one.

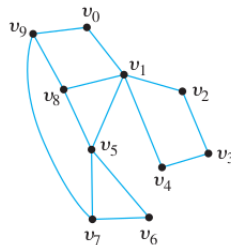
### 1.12 Exercise 12



*Proof.* One Euler circuit is  $e_4e_5e_6e_3e_2e_7e_8e_1$ .

□

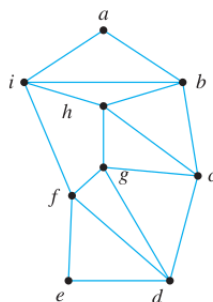
### 1.13 Exercise 13



*Proof.* No Euler circuit, since  $v_1$  has odd degree (5).

□

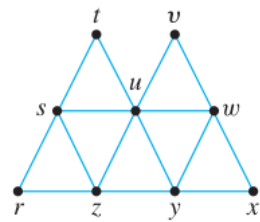
### 1.14 Exercise 14



*Proof.* One Euler circuit is  $iabihbchgcdgfdedefi$ .

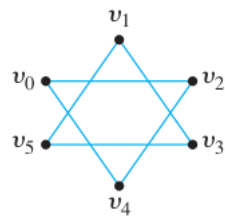
□

1.15    Exercise 15



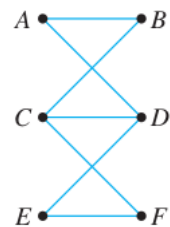
*Proof.* One Euler circuit is  $rzyxwyuzsuwvutsr$ . □

1.16    Exercise 16



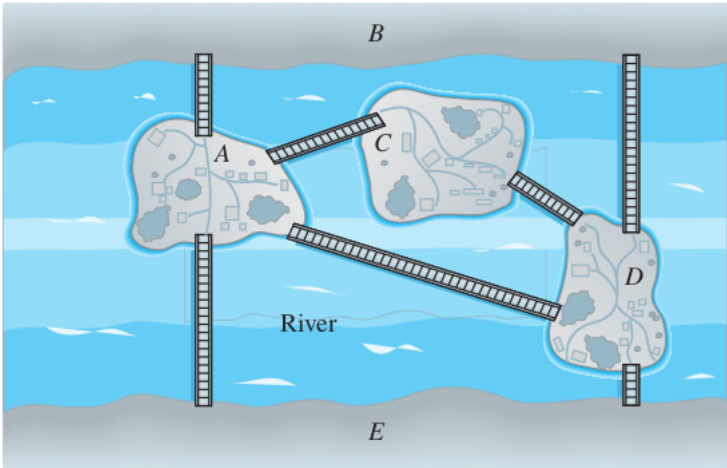
*Proof.* No Euler circuit (graph is not connected). □

1.17    Exercise 17



*Proof.* No Euler circuit (C has odd degree 3). □

1.18    Exercise 18

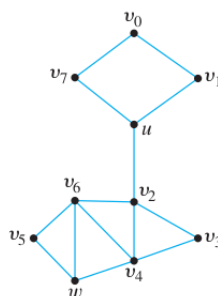


Is it possible to take a walk around the city whose map is shown above, starting and ending at the same point and crossing each bridge exactly once? If so, how can this be done?

*Proof.* DCADBAED □

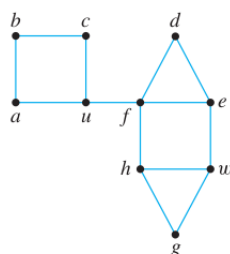
For each of the graphs in 19 – 21, determine whether there is an Euler trail from  $u$  to  $w$ . If there is, find such a trail.

## 1.19 Exercise 19



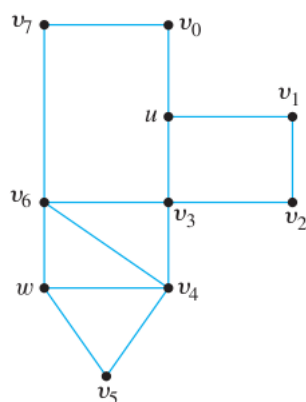
*Proof.* There is an Euler trail since  $\deg(u)$  and  $\deg(w)$  are odd, all other vertices have positive even degree, and the graph is connected.  $uv_1v_0v_7uv_2v_3v_4v_2v_6v_4wv_5v_6w$  is one Euler trail. □

## 1.20 Exercise 20



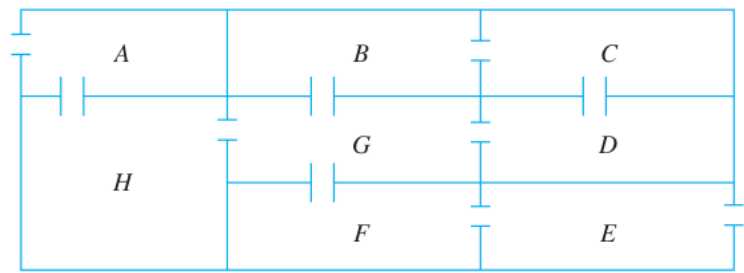
*Proof.* No Euler trail, since there are 4 vertices with odd degrees:  $u, e, h, w$ . □

## 1.21 Exercise 21



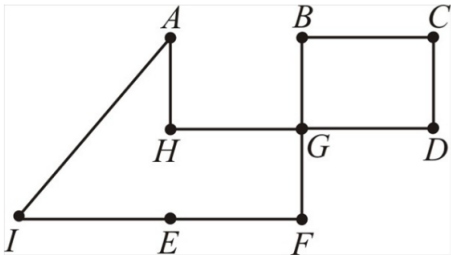
*Proof.* One Euler trail is  $uv_0v_7v_6v_3uv_1v_2v_3v_4v_6wv_4v_5w$ . □

### 1.22 Exercise 22



The following is a floor plan of a house. Is it possible to enter the house in room A, travel through every interior doorway of the house exactly once, and exit out of room E? If so, how can this be done?

*Proof.* Let rooms be vertices, let the outside also be a vertex (I). Then

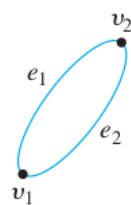


One Euler circuit is: IAHGBCDGF EI □

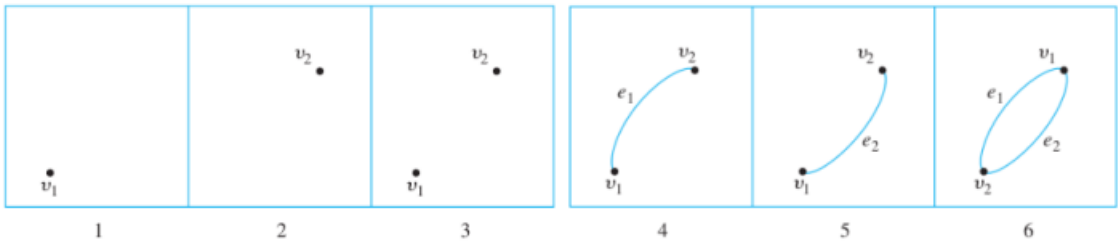
### 1.23 Exercise 23

Find all subgraphs of each of the following graphs.

#### 1.23.1 (a)

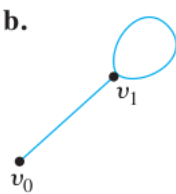


*Proof.* The nonempty subgraphs are as follows:

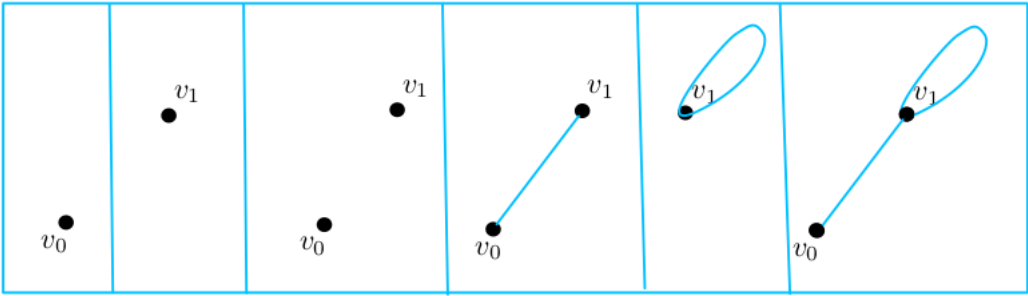


□

1.23.2 (b)

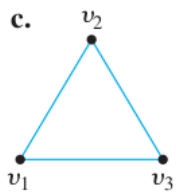


*Proof.* The nonempty subgraphs are as follows:

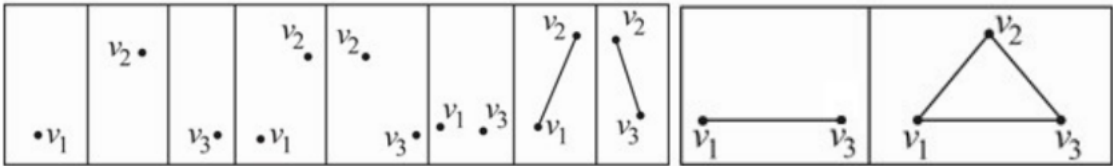


□

1.23.3 (c)

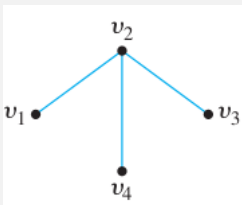


*Proof.* The nonempty subgraphs are as follows:



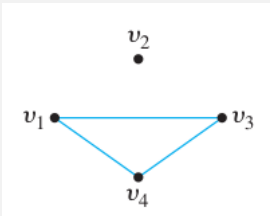
□

**Definition:** If  $G$  is a simple graph, the complement of  $G$ , denoted  $G'$ , is obtained as follows: The vertex set of  $G'$  is identical to the vertex set of  $G$ . However, two distinct vertices  $v$  and  $w$  of  $G'$  are connected by an edge if, and only if,  $v$  and  $w$  are not connected by an edge in  $G$ .



For example, if  $G$  is the graph

then  $G'$  is the graph

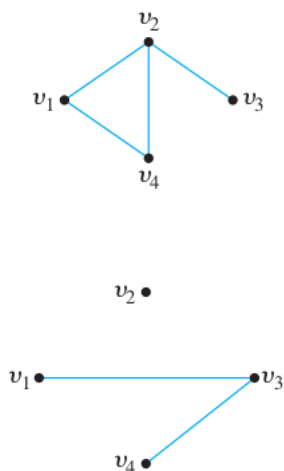


.

## 1.24 Exercise 24

Find the complement of each of the following graphs.

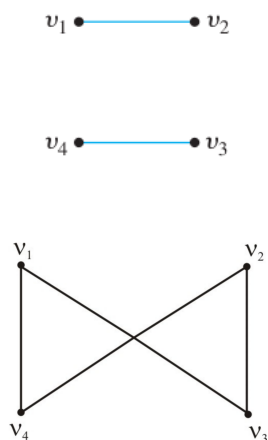
### 1.24.1 (a)



*Proof.*

□

### 1.24.2 (b)



*Proof.*

□

## 1.25 Exercise 25

### 1.25.1 (a)

Find the complement of the graph  $K_4$ , the complete graph on four vertices.

*Proof.* This is the graph with four vertices and no edges.

□

### 1.25.2 (b)

Find the complement of the graph  $K_{3,2}$ , the complete bipartite graph on  $(3, 2)$  vertices.

*Proof.* This is the union of separate graphs  $K_3$  (a triangle) and  $K_2$  (a line).

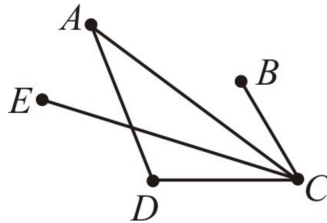
□

## 1.26 Exercise 26

Suppose that in a group of five people A, B, C, D, and E the following pairs of people are acquainted with each other. A and C, A and D, B and C, C and D, C and E.

### 1.26.1 (a)

Draw a graph to represent this situation.

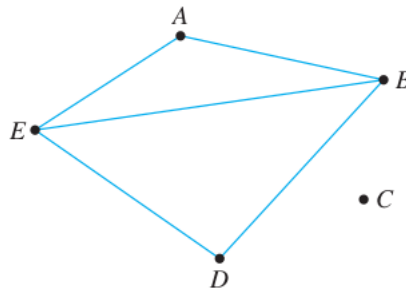


*Proof.*

□

### 1.26.2 (b)

Draw a graph that illustrates who among these five people are not acquainted. That is, draw an edge between two people if, and only if, they are not acquainted.



*Proof.*

□

## 1.27 Exercise 27

Let  $G$  be a simple graph with  $n$  vertices. What is the relation between the number of edges of  $G$  and the number of edges of the complement  $G'$ ?

*Hint:* Consider the graph obtained by taking the vertices and edges of  $G$  plus all the edges of  $G'$ .

*Proof.*  $G'$  has all the edges that  $G$  does not between any two vertices. Therefore, if we put all the edges of  $G$  and  $G'$  together, it should give us all possible edges between any two vertices, in other words, the complete graph  $K_n$  on  $n$  vertices. We know  $K_n$  has  $\frac{n(n-1)}{2}$  edges total, therefore: “the number of edges of  $G$  plus the number of edges of  $G'$  equals  $\frac{n(n-1)}{2}$ .” □

## 1.28 Exercise 28

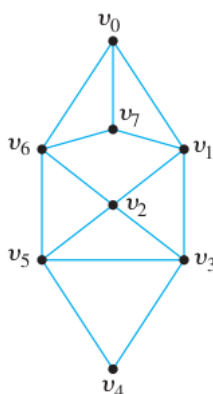
Show that at a party with at least two people, there are at least two mutual acquaintances or at least two mutual strangers.

*Proof.* Since there are at least 2 people at the party, consider person 1 and person 2. Either they know each other (so they are mutually acquainted), or they do not (so they are mutual strangers).

(This problem is stated weirdly! I have to assume that acquaintance is a symmetric relation, given Exercise 26 above: if person A is acquainted with person B, then person B is also acquainted with person A; in other words A cannot know B non-mutually without B knowing A. Otherwise, the acquaintance graph would have to be a directed graph, which has not been yet covered in this section.)  $\square$

Find Hamiltonian circuits for each of the graphs in 29 and 30.

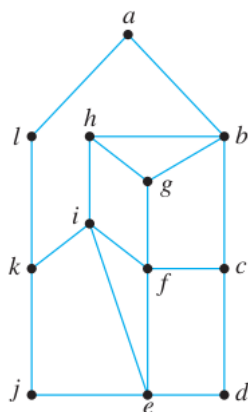
## 1.29 Exercise 29



*Proof.*  $v_0v_7v_1v_2v_3v_4v_5v_6v_0$

$\square$

## 1.30 Exercise 30



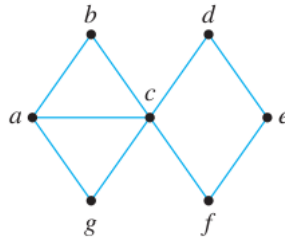
*Proof.*  $alkjedcfihgba$

$\square$

Show that none of the graphs in 31 – 33 has a Hamiltonian circuit.

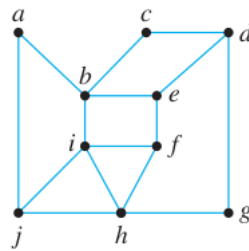


### 1.31 Exercise 31



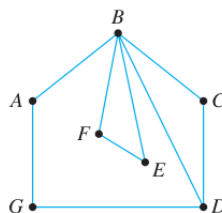
*Proof.* Argue by contradiction and assume  $G$  has a Hamiltonian circuit. By Proposition 10.1.6  $G$  has a subgraph  $H$  that is connected, contains every vertex of  $G$ , has the same number of edges as vertices, and every vertex has degree 2. Since vertex  $c$  has degree 5, in the subgraph  $H$  3 of these edges must be removed. But 4 of these 5 edges cannot be removed, otherwise  $b, d, g, f$  would have degree less than 2. This is a contradiction! So  $G$  has no Hamiltonian circuit.  $\square$

### 1.32 Exercise 32



*Proof.* Call the given graph  $G$ , and suppose  $G$  has a Hamiltonian circuit. Then  $G$  has a subgraph  $H$  that satisfies conditions (1)–(4) of Proposition 10.1.6. Since the degree of  $b$  in  $G$  is 4 and every vertex in  $H$  has degree 2, two edges incident on  $b$  must be removed from  $G$  to create  $H$ . Edge  $\{a, b\}$  cannot be removed because doing so would result in vertex  $d$  having degree less than 2 in  $H$ . Similar reasoning shows that edge  $\{b, c\}$  cannot be removed either. So edges  $\{b, i\}$  and  $\{b, e\}$  must be removed from  $G$  to create  $H$ . Because vertex  $e$  must have degree 2 in  $H$  and because edge  $\{b, e\}$  is not in  $H$ , both edges  $\{e, d\}$  and  $\{e, f\}$  must be in  $H$ . Similarly, since both vertices  $c$  and  $g$  must have degree 2 in  $H$ , edges  $\{c, d\}$  and  $\{g, d\}$  must also be in  $H$ . But then three edges incident on  $d$ , namely,  $\{e, d\}$ ,  $\{c, d\}$ , and  $\{g, d\}$ , must all be in  $H$ , which contradicts the fact that vertex  $d$  must have degree 2 in  $H$ .  $\square$

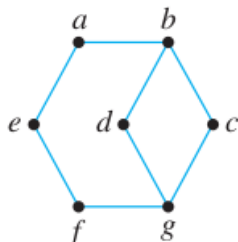
### 1.33 Exercise 33



*Proof.* Similar argument to Exercises 31, 32 above. In  $H$  every vertex has degree 2. Since  $B$  has degree 5, 3 edges must be removed, but 4 of these 5 edges cannot be removed, otherwise  $A, F, E, C$  would have degree less than 2.  $\square$

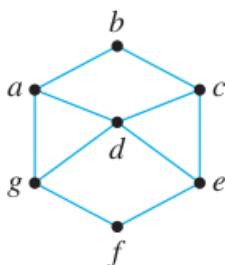
In 34–37, find Hamiltonian circuits for those graphs that have them. Explain why the other graphs do not.

### 1.34 Exercise 34



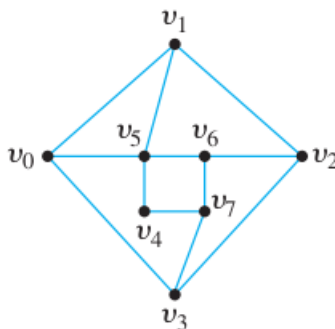
*Proof.* This graph does not have a Hamiltonian circuit. Argue like Exercises 31, 32, 33 above. In  $H$  every vertex has degree 2. Since  $b$  has degree 3, it must have one edge removed in  $H$ . But none of the 3 edges incident on  $b$  can be removed: otherwise one of  $a, d, c$  would have degree less than 2.  $\square$

### 1.35 Exercise 35

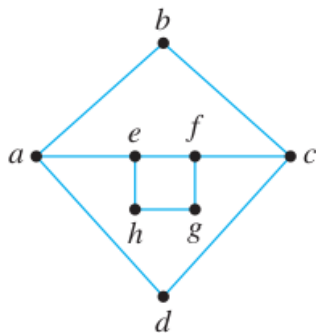


*Proof.*  $adgfe cba$   $\square$

### 1.36 Exercise 36



*Proof.*  $v_4v_5v_1v_0v_3v_2v_6v_7v_4$   $\square$



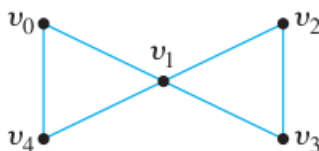
### 1.37 Exercise 37

*Proof.* This graph has no Hamiltonian circuits. Similar argument as before: in  $H$  every vertex has degree 2. Since  $a$  has degree 3, we must remove one edge. That has to be  $\{a, e\}$  otherwise  $b$  or  $d$  would have degree 1. Similarly  $c$  has degree 3, we must remove one edge, it has to be  $\{f, c\}$  otherwise  $b$  or  $d$  would have degree 1. But after removing those 2 edges, we are left with a disconnected graph: a “diamond” ( $a - b - c - d$ ) with a “square” inside it ( $e - f - g - h$ ), which contradicts the fact that  $H$  must be connected.  $\square$

### 1.38 Exercise 38

Give two examples of graphs that have Euler circuits but not Hamiltonian circuits.

*Proof.* One example: This graph has an Euler circuit  $v_0v_1v_2v_3v_1v_4v_0$  but no Hamiltonian circuit.

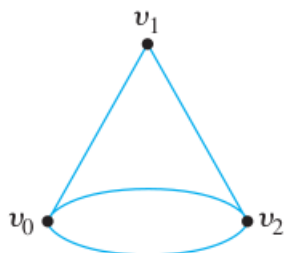


$\square$

### 1.39 Exercise 39

Give two examples of graphs that have Hamiltonian circuits but not Euler circuits.

*Proof.* One example: This graph has a Hamiltonian circuit  $v_0v_1v_2v_0$  but no Euler circuit.

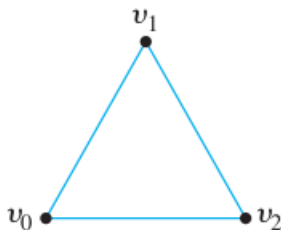


$\square$

### 1.40 Exercise 40

Give two examples of graphs that have circuits that are both Euler circuits and Hamiltonian circuits.

*Proof.* One example: The walk  $v_0v_1v_2v_0$  is both an Euler circuit and a Hamiltonian circuit for this graph.

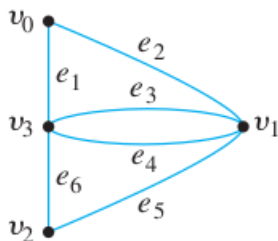


□

### 1.41 Exercise 41

Give two examples of graphs that have Euler circuits and Hamiltonian circuits that are not the same.

*Proof.* One example: This graph has the Euler circuit  $e_1e_2e_3e_4e_5e_6$  and the Hamiltonian circuit  $v_0v_1v_2v_3v_0$ . These are not the same.



□

### 1.42 Exercise 42

A traveler in Europe wants to visit each of the cities shown on the map exactly once, starting and ending in Brussels. The distance (in kilometers) between each pair of cities is given in the table. Find a Hamiltonian circuit that minimizes the total distance traveled. (Use the map to narrow the possible circuits down to just a few. Then use the table to find the total distance for each of those.)

*Proof.* Find a Hamiltonian circuit that minimizes the total distance traveled. Some possible Hamiltonian circuits are as follows:

1. Brussels-Luxembourg-Dusseldorf-Paris-Munich-Berlin-Brussels. Total distance traveled by the visitor is  $219 + 224 + 497 + 832 + 585 + 783 = 3140$  km.



	Berlin	Brussels	Düsseldorf	Luxembourg	Munich
Brussels	783				
Düsseldorf	564	223			
Luxembourg	764	219	224		
Munich	585	771	613	517	
Paris	1,057	308	497	375	832

2. Brussels-Luxembourg-Paris-Munich-Berlin-Dusseldorf-Brussels. Total distance traveled by the visitor is  $219 + 375 + 832 + 585 + 564 + 223 = 2798$  km.

3. Brussels-Paris-Luxembourg-Munich-Berlin-Dusseldorf-Brussels. Total distance traveled by the visitor is  $308 + 375 + 517 + 585 + 564 + 223 = 2572$  km.

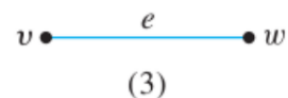
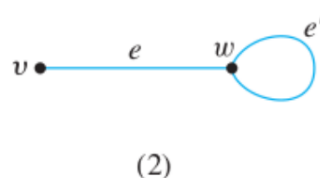
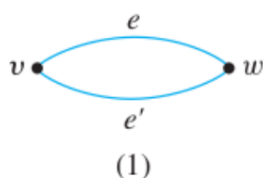
Therefore, the Hamiltonian circuit is Brussels-Paris-Luxembourg-Munich-Berlin-Dusseldorf-Brussels in which total distance is minimum.  $\square$

## 1.43 Exercise 43

### 1.43.1 (a)

Prove that if a walk in a graph contains a repeated edge, then the walk contains a repeated vertex.

*Proof.* Suppose  $G$  is a graph and  $W$  is a walk in  $G$  that contains a repeated edge  $e$ . Let  $v$  and  $w$  be the endpoints of  $e$ . In case  $v = w$ , then  $v$  is a repeated vertex of  $W$ . In case  $v \neq w$ , then one of the following must occur: (1)  $W$  contains two copies of  $vev$  or of  $wew$  (for instance,  $W$  might contain a section of the form  $veve'vew$ , as illustrated below);



(2)  $W$  contains separate sections of the form  $vev$  and  $wew$  (for instance,  $W$  might contain a section of the form  $veve'wew$ , as illustrated below); or (3)  $W$  contains a section of the form  $vevew$  or of the form  $wewew$  (as illustrated below). In cases (1) and (2), both vertices  $v$  and  $w$  are repeated, and in case (3), one of  $v$  or  $w$  is repeated. In all cases, there is at least one vertex in  $W$  that is repeated.  $\square$

### 1.43.2 (b)

Explain how it follows from part (a) that any walk with no repeated vertex has no repeated edge.

*Proof.* This statement is the contrapositive of part (a).  $\square$

## 1.44 Exercise 44

Prove Lemma 10.1.1(a): If  $G$  is a connected graph, then any two distinct vertices of  $G$  can be connected by a path. (You may use the result stated in exercise 43.)

*Proof.* Suppose  $G$  is a connected graph and  $v$  and  $w$  are any particular but arbitrarily chosen vertices of  $G$ . [We must show that  $u$  and  $v$  can be connected by a path.] Since  $G$  is connected, there is a walk from  $v$  to  $w$ . If the walk contains a repeated vertex, then delete the portion of the walk from the first occurrence of the vertex to its next occurrence. (For example, in the walk  $ve_1v_2e_5v_7e_6v_2e_3w$ , the vertex  $v_2$  occurs twice. Deleting the portion of the walk from one occurrence to the next gives  $ve_1v_2e_3w$ .) If the resulting walk still contains a repeated vertex, do the above deletion process another time. Then check again for a repeated vertex. Continue in this way until all repeated vertices have been deleted. (This must occur eventually, since the total number of vertices is finite.) The resulting walk connects  $v$  to  $w$  but has no repeated vertex. By exercise 43(b), it has no repeated edge either. Hence it is a path from  $v$  to  $w$ .  $\square$

## 1.45 Exercise 45

Prove Lemma 10.1.1(b): If vertices  $v$  and  $w$  are part of a circuit in a graph  $G$  and one edge is removed from the circuit, then there still exists a trail from  $v$  to  $w$  in  $G$ .

*Proof.* Write down the circuit as:

$$ve_1v_1e_2v_2 \dots v_ne_nv_1f_1w_1f_2w_2 \dots f_{m-1}w_{m-1}f_mv$$

If one of the edges  $e_1, \dots, e_n$  is removed then there is still a trail

$$wf_1w_1f_2w_2 \dots f_{m-1}w_{m-1}f_mv$$

between  $w$  and  $v$ , written backwards it's a trail from  $v$  to  $w$ , and similarly if one of the edges  $f_1, \dots, f_m$  is removed then there is still a trail

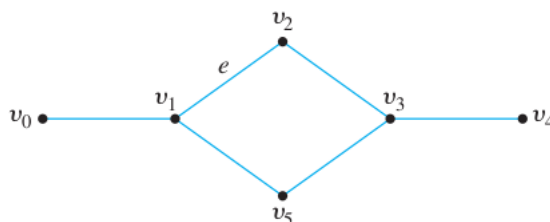
$$ve_1v_1e_2v_2 \dots v_ne_nv_1w$$

from  $v$  to  $w$ .  $\square$

## 1.46 Exercise 46

Draw a picture to illustrate Lemma 10.1.1(c): If a graph  $G$  is connected and  $G$  contains a circuit, then an edge of the circuit can be removed without disconnecting  $G$ .

*Proof.* The graph below contains a circuit, any edge of which can be removed without disconnecting the graph. For instance, if edge  $e$  is removed, then the following walk can be used to go from  $v_1$  to  $v_2$ :  $v_1v_5v_3v_2$ .



□

## 1.47 Exercise 47

Prove that if there is a trail in a graph  $G$  from a vertex  $v$  to a vertex  $w$ , then there is a trail from  $w$  to  $v$ .

*Proof.* Simply traverse the trail from  $v$  to  $w$  but starting at  $w$  and going backwards. This is also a trail from  $w$  to  $v$  (no repeated edges). □

## 1.48 Exercise 48

If a graph contains a circuit that starts and ends at a vertex  $v$ , does the graph contain a simple circuit that starts and ends at  $v$ ? Why?

*Proof.* Yes, suppose a graph contains a circuit that starts and ends at a vertex  $v$ . Successively delete sections of this circuit as follows. For each repeated vertex  $w$  in the circuit (excluding the first vertex if its only repetition is at the end of the circuit, but including the first vertex if it is repeated in the middle of the circuit), there is a section of the circuit of the form  $we_1v_1 \dots e_{n-1}v_{n-1}e_nv$ . Replace this section of the circuit by the single vertex  $w$ . Because the circuit has finite length, only a finite number of such deletions can be made, after which a simple circuit starting and ending at  $v$  will remain. □

## 1.49 Exercise 49

Prove that if there is a circuit in a graph that starts and ends at a vertex  $v$  and if  $w$  is another vertex in the circuit, then there is a circuit in the graph that starts and ends at  $w$ .

*Proof.* If a circuit starts and ends at  $v$  and contains  $w$  then the circuit has the form

$$ve_1v_1e_2v_2 \dots v_ne_nwf_1w_1f_2w_2 \dots f_{m-1}w_{m-1}f_mv.$$

Then the following is a circuit that starts and ends at  $w$ :

$$wf_1w_1f_2w_2\cdots f_{m-1}w_{m-1}f_mv_e1v_1e_2v_2\cdots v_ne_nw$$

□

## 1.50 Exercise 50

Let  $G$  be a connected graph, and let  $C$  be any circuit in  $G$  that does not contain every vertex of  $G$ . Let  $G'$  be the subgraph obtained by removing all the edges of  $C$  from  $G$  and also any vertices that become isolated when the edges of  $C$  are removed. Prove that there exists a vertex  $v$  such that  $v$  is in both  $C$  and  $G'$ .

*Proof.* Let  $G$  be a connected graph and let  $C$  be a circuit in  $G$ . Let  $G'$  be the subgraph obtained by removing all the edges of  $C$  from  $G$  and also any vertices that become isolated when the edges of  $C$  are removed. [We must show that there exists a vertex  $v$  such that  $v$  is in both  $C$  and  $G'$ .] Pick any vertex  $v$  of  $C$  and any vertex  $w$  of  $G'$ . Since  $G$  is connected, there is a path from  $v$  to  $w$  (by Lemma 10.1.1(a)):

$$\begin{array}{cccccccccccccccc} v & = & v_0 & e_1 & v_1 & \cdots & e_i & v_i & e_{i+1} & v_{i+1} & \cdots & v_{n-1} & e_n & v_n & = & w \\ \uparrow & & & & & & & \uparrow & & \uparrow & & & & & & \uparrow \\ \text{in } C & & & & & & & \text{in } C & & \text{not in } C & & & & & & \text{in } G' \end{array}$$

Let  $i$  be the largest subscript such that  $v_i$  is in  $C$ . If  $i = n$ , then  $v_n = w$  is in  $C$  and also in  $G'$ , and we are done. If  $i < n$ , then  $v_i$  is in  $C$  and  $v_{i+1}$  is not in  $C$ . This implies that  $e_{i+1}$  is not in  $C$  (for if it were, both endpoints would be in  $C$  by definition of circuit). Hence when  $G'$  is formed by removing the edges and resulting isolated vertices from  $G$ , then  $e_{i+1}$  is not removed. That means that  $v_i$  does not become an isolated vertex, so  $v_i$  is not removed either. Hence  $v_i$  is in  $G'$ . Consequently,  $v_i$  is in both  $C$  and  $G'$  [as was to be shown]. □

## 1.51 Exercise 51

Prove that any graph with an Euler circuit is connected.

*Proof.* Suppose  $G$  is a graph with an Euler circuit. If  $G$  has only one vertex, then  $G$  is automatically connected. If  $v$  and  $w$  are any two vertices of  $G$ , then  $v$  and  $w$  each appear at least once in the Euler circuit (since an Euler circuit contains every vertex of the graph). The section of the circuit between the first occurrence of one of  $v$  or  $w$  and the first occurrence of the other is a walk from one of the two vertices to the other. Since the choice of  $v$  and  $w$  was arbitrary, given any two vertices in  $G$  there is a walk from one to the other. So, by definition,  $G$  is connected. □

## 1.52 Exercise 52

Prove Corollary 10.1.5.



*Corollary:* Let  $G$  be a graph, and let  $v$  and  $w$  be two distinct vertices of  $G$ . There is an Euler trail from  $v$  to  $w$  if, and only if,  $G$  is connected,  $v$  and  $w$  have odd degree, and all other vertices of  $G$  have positive even degree.

*Proof.* (  $\implies$  ) Assume there is an Euler trail from  $v$  to  $w$ .

Then this trail passes through every vertex of  $G$  at least once and traverses every edge of  $G$  exactly once, therefore  $G$  is connected. Adding an extra edge  $e$  between  $v$  and  $w$  to this trail gives us an Euler circuit. Then every vertex in this new graph has even degree. Removing  $e$  shows that  $v$  and  $w$  have odd degree and all other vertices have even degree in the original graph  $G$ .

(  $\impliedby$  ) Assume  $G$  is connected,  $v$  and  $w$  have odd degree, and all other vertices of  $G$  have positive even degree.

Add an extra edge  $e$  between  $v$  and  $w$ . Now every vertex in this new graph has even degree, and it is connected. So it has an Euler circuit by Theorem 10.1.4. This Euler circuit contains  $e$ . Removing  $e$  from the circuit gives us an Euler trail from  $v$  to  $w$  in the original graph  $G$ .  $\square$

### 1.53 Exercise 53

For what values of  $n$  does the complete graph  $K_n$  with  $n$  vertices have (a) an Euler circuit? (b) a Hamiltonian circuit? Justify your answers.

*Proof.*  $K_n$  has a circuit if and only if  $n \geq 3$ .

(a) Euler circuits require a connected graph where every vertex has even degree. Every vertex in  $K_n$  has degree  $n - 1$ . So this means that for all  $n$ ,  $K_n$  has an Euler circuit if and only if  $n \geq 3$  and  $n$  is odd.

(b) Hamiltonian circuits require a connected subgraph  $H$  with the same number of edges and vertices, which contains every vertex of  $G$ , such that every vertex has degree 2 in  $H$ . This is possible in every  $K_n$  with  $n \geq 3$ . If the vertices are  $v_1, \dots, v_n$  then let  $H$  be  $v_1v_2 \cdots v_{n-1}v_nv_1$ . This is a connected circuit with  $n$  vertices and  $n$  edges, where every vertex has degree 2.  $\square$

### 1.54 Exercise 54

For what values of  $m$  and  $n$  does the complete bipartite graph on  $(m, n)$  vertices have (a) an Euler circuit? (b) a Hamiltonian circuit? Justify your answers.

*Proof.* (a) Euler circuit requires every vertex to have even degree, and in  $K_{m,n}$  the left  $m$  vertices all have degree  $n$  and the right  $n$  vertices all have degree  $m$ . So  $K_{m,n}$  has an Euler circuit if and only if both  $m$  and  $n$  are even (and  $\geq 2$ ).

(b) Hamiltonian circuit requires every vertex to have degree 2. The total degree of the left  $m$  vertices is  $2m$ , so there are  $2m$  edges going from the left  $m$  vertices to the right

$n$  vertices, covering a total degree of  $2n$  on the right. This forces  $2m = 2n$  so  $m = n$ . So  $K_{m,n}$  has a Hamiltonian circuit if and only if  $m = n$  (and  $\geq 2$ ).  $\square$

## 1.55 Exercise 55

What is the maximum number of edges a simple disconnected graph with  $n$  vertices can have? Prove your answer.

*Proof.* Simple means no loops or parallel edges. Leave out 1 vertex by itself so it's not connected to any of the other  $n - 1$ . Now we try to find the maximum number of edges we can have among the  $n - 1$  vertices without introducing any loops. We can add  $n - 2$  edges between them in a chain. Adding 1 more edge to this would create a loop. So the maximum is  $n - 2$ .  $\square$

## 1.56 Exercise 56

### 1.56.1 (a)

Prove that if  $G$  is any bipartite graph, then every circuit in  $G$  has an even number of edges.

*Proof.* Let  $G = (V_1, V_2)$  be a bipartite graph where  $V_1$  is the set of vertices on the left and  $V_2$  is the set of vertices on the right. Consider any circuit. Say, it starts and ends at a vertex  $v \in V_1$ . Since  $v$  can only be connected to vertices in  $V_2$ , one edge brings us to  $V_2$ . Similarly, since vertices in  $V_2$  can only be connected to vertices in  $V_1$ , the second edge of the circuit brings us back to  $V_1$ . And so on. Any odd number of edges starting from  $v$  will end in  $V_2$ , unable to complete the circuit back at  $v$ . Hence, no circuit can have an odd number of edges, but must have an even number of edges.  $\square$

### 1.56.2 (b)

Prove that if  $G$  is any graph with at least two vertices and if  $G$  does not have a circuit with an odd number of edges, then  $G$  is bipartite.

*Hint:* Divide the proof into three parts. (1) Show that if  $G$  is any graph containing a closed walk with an odd number of edges, then  $G$  contains a circuit with an odd number of edges. (2) Show that if  $G$  is any connected graph that does not have a circuit with an odd number of edges, then  $G$  is bipartite. (3) Show that if  $G$  is any graph with at least two vertices and is such that  $G$  does not have a circuit with an odd number of edges, then  $G$  is bipartite.

*Proof.* (1) Let  $W$  be a closed walk with an odd number of edges. If no vertices are repeated in  $W$  then  $W$  is already a circuit with an odd number of edges. So suppose  $W$  has a repeated vertex. Write  $W = v_1v_2 \cdots v_jv_{j+1} \cdots v_kv_{k+1} \cdots v_nv_1$  where  $v_j = v_k$  is the repeated vertex. Now  $W$  is the union of two closed walks:  $v_1v_2 \cdots v_jv_{k+1} \cdots v_nv_1$  and  $v_jv_{j+1} \cdots v_kv_k$ . Since the number of edges of  $W$  is odd, and the sum of the numbers of edges of these two walks equals the number of edges of  $W$ , one of these two walks must

have an odd number of edges and the other must have an even number of edges (because  $\text{odd} + \text{odd} = \text{even}$ ,  $\text{even} + \text{even} = \text{even}$ , and only  $\text{odd} + \text{even} = \text{odd}$ ). Take the walk among the two walks that has the odd number of edges. Either this smaller walk is a circuit, or it also has a repeated vertex. Then repeat the procedure to obtain an even smaller closed walk with an odd number of edges. The process eventually terminates since  $W$  is finite. So there is a circuit with an odd number of edges.

(2) Assume  $G$  is connected and has no circuit with an odd number of edges. Let  $v$  be a vertex in  $G$ . Since  $G$  is connected, there is a walk between  $v$  and any other vertex. For any vertex  $w$  let  $d(w, v)$  denote the length of the shortest walk between  $v$  and  $w$ . Let's define a bipartition of  $G$  as follows:

$$V_1 = \{w \in V(G) \mid d(w, v) \text{ is odd}\}, \quad V_2 = \{w \in V(G) \mid d(w, v) \text{ is even}\}$$

We need to prove  $(V_1, V_2)$  is a bipartition of  $G$ . Argue by contradiction and assume  $x, y \in V_1$  and assume there is an edge  $e$  between  $x$  and  $y$ . Now there is a closed walk with an odd number of edges from  $v$  to  $v$  as follows: start with a walk from  $v$  to  $x$  which has odd length (since  $x \in V_1$ ), followed by  $e$  (add 1), followed by the walk from  $y$  to  $v$  which has odd length. This is a closed walk of odd length, which, by (1), implies there is a circuit of odd length from  $v$  to  $v$ , which is a contradiction to the assumption that  $G$  has no circuits of odd length. Similar argument for  $V_2$ .

(3) Assume  $G$  is any graph with at least two vertices and has no circuit with an odd number of edges. Consider all the connected components  $G_1, \dots, G_n$  of  $G$ . By (2) each  $G_i$  is bipartite:  $G_i = (V_{1,i}, V_{2,i})$ . Then we can create a bipartition of  $G$ :  $(\bigcup_{i=1}^n V_{1,i}, \bigcup_{i=1}^n V_{2,i})$ . So  $G$  is bipartite.  $\square$

## 1.57 Exercise 57

An alternative proof for Theorem 10.1.3 has the following outline. Suppose  $G$  is a connected graph in which every vertex has even degree. Suppose the trail

$C : v_1 e_1 v_2 e_2 v_3 \dots e_n v_{n+1}$  has maximum length in  $G$ . That is,  $C$  has at least as many vertices and edges as any other trail in  $G$ . First derive a contradiction from the assumption that  $v_1 \neq v_{n+1}$ . Next let  $H$  be the subgraph of  $G$  that contains all the vertices and edges in  $C$ . Then derive a contradiction from the assumption that  $H \neq G$ . Show that  $H$  contains every vertex of  $G$ , and show that  $H$  contains every edge of  $G$ .

*Proof.* Argue by contradiction and assume  $v_1 \neq v_{n+1}$ . Note that  $v_1$  is connected to only one other vertex,  $v_2$ , on  $C$ , and  $v_{n+1}$  is connected to only one other vertex,  $v_n$ , in  $C$ . Since both  $v_1$  and  $v_{n+1}$  have even degree, they have degree at least 2. So they are connected to at least one more vertex. Then there is a trail in  $G$  longer than  $C$ , contradiction. So  $v_1 = v_{n+1}$ .

Let  $H$  be the subgraph of  $G$  that contains all the vertices and edges in  $C$ . Argue by contradiction and assume  $H \neq G$ . First assume  $G$  has a vertex  $w$  that  $H$  does not. Since  $w$  is different than all  $v_1, \dots, v_n$ ,  $w$  is not on  $C$ . Since  $G$  is connected there is a trail from  $v_1$  to  $w$ . Since  $w$  is not on  $C$ , this trail is not part of  $C$ . Adding this trail to  $C$  gives us a longer trail, contradiction. So  $H$  contains every vertex of  $G$ .

Now assume  $G$  has an edge  $e$  that  $H$  does not. This edge must be connecting two of the vertices  $v_1, \dots, v_n$ . Say  $v_i$  and  $v_j$ . But then  $v_i$  and  $v_j$  have degree 3, contradiction. So  $H$  contains every edge of  $G$ . □

## 2 Exercise Set 10.2

### 2.1 Exercise 1

2.1.1 ()

*Proof.*

□

### 2.2 Exercise 2

2.2.1 ()

*Proof.*

□

### 2.3 Exercise 3

2.3.1 ()

*Proof.*

□

### 2.4 Exercise 4

2.4.1 ()

*Proof.*

□

### 2.5 Exercise 5

2.5.1 ()

*Proof.*

□

### 2.6 Exercise 6

2.6.1 ()

*Proof.*

□

### 2.7 Exercise 7

2.7.1 ()

*Proof.*

□

## 2.8 Exercise 8

### 2.8.1 ()

*Proof.*



## 2.9 Exercise 9

### 2.9.1 ()

*Proof.*



## 2.10 Exercise 10

### 2.10.1 ()

*Proof.*



## 2.11 Exercise 11

### 2.11.1 ()

*Proof.*



## 2.12 Exercise 12

### 2.12.1 ()

*Proof.*



## 2.13 Exercise 13

### 2.13.1 ()

*Proof.*



## 2.14 Exercise 14

### 2.14.1 ()

*Proof.*



## 2.15 Exercise 15

### 2.15.1 ()

*Proof.*



## **2.16 Exercise 16**

**2.16.1**     $()$

*Proof.*



## **2.17 Exercise 17**

**2.17.1**     $()$

*Proof.*



## **2.18 Exercise 18**

**2.18.1**     $()$

*Proof.*



## **2.19 Exercise 19**

**2.19.1**     $()$

*Proof.*



## **2.20 Exercise 20**

**2.20.1**     $()$

*Proof.*



## **2.21 Exercise 21**

**2.21.1**     $()$

*Proof.*



## **2.22 Exercise 22**

**2.22.1**     $()$

*Proof.*



## **2.23 Exercise 23**

**2.23.1**     $()$

*Proof.*



### 3 Exercise Set 10.3

#### 3.1 Exercise 1

3.1.1 ()

*Proof.*

□

#### 3.2 Exercise 2

3.2.1 ()

*Proof.*

□

#### 3.3 Exercise 3

3.3.1 ()

*Proof.*

□

#### 3.4 Exercise 4

3.4.1 ()

*Proof.*

□

#### 3.5 Exercise 5

3.5.1 ()

*Proof.*

□

#### 3.6 Exercise 6

3.6.1 ()

*Proof.*

□

#### 3.7 Exercise 7

3.7.1 ()

*Proof.*

□

#### 3.8 Exercise 8

3.8.1 ()

*Proof.*

□

### **3.9 Exercise 9**

#### **3.9.1 ()**

*Proof.*

□

### **3.10 Exercise 10**

#### **3.10.1 ()**

*Proof.*

□

### **3.11 Exercise 11**

#### **3.11.1 ()**

*Proof.*

□

### **3.12 Exercise 12**

#### **3.12.1 ()**

*Proof.*

□

### **3.13 Exercise 13**

#### **3.13.1 ()**

*Proof.*

□

### **3.14 Exercise 14**

#### **3.14.1 ()**

*Proof.*

□

### **3.15 Exercise 15**

#### **3.15.1 ()**

*Proof.*

□

### **3.16 Exercise 16**

#### **3.16.1 ()**

*Proof.*

□



### **3.17 Exercise 17**

**3.17.1**    ()

*Proof.*

□

### **3.18 Exercise 18**

**3.18.1**    ()

*Proof.*

□

### **3.19 Exercise 19**

**3.19.1**    ()

*Proof.*

□

### **3.20 Exercise 20**

**3.20.1**    ()

*Proof.*

□

### **3.21 Exercise 21**

**3.21.1**    ()

*Proof.*

□

### **3.22 Exercise 22**

**3.22.1**    ()

*Proof.*

□

### **3.23 Exercise 23**

**3.23.1**    ()

*Proof.*

□

### **3.24 Exercise 24**

**3.24.1**    ()

*Proof.*

□

### **3.25 Exercise 25**

**3.25.1** ()

*Proof.*

☐

### **3.26 Exercise 26**

**3.26.1** ()

*Proof.*

☐

### **3.27 Exercise 27**

**3.27.1** ()

*Proof.*

☐

### **3.28 Exercise 28**

**3.28.1** ()

*Proof.*

☐

### **3.29 Exercise 29**

**3.29.1** ()

*Proof.*

☐

### **3.30 Exercise 30**

**3.30.1** ()

*Proof.*

☐

## **4 Exercise Set 10.4**

### **4.1 Exercise 1**

**4.1.1** ()

*Proof.*

☐

### **4.2 Exercise 2**

**4.2.1** ()

*Proof.*

☐

### 4.3 Exercise 3

4.3.1 ()

*Proof.*

□

### 4.4 Exercise 4

4.4.1 ()

*Proof.*

□

### 4.5 Exercise 5

4.5.1 ()

*Proof.*

□

### 4.6 Exercise 6

4.6.1 ()

*Proof.*

□

### 4.7 Exercise 7

4.7.1 ()

*Proof.*

□

### 4.8 Exercise 8

4.8.1 ()

*Proof.*

□

### 4.9 Exercise 9

4.9.1 ()

*Proof.*

□

### 4.10 Exercise 10

4.10.1 ()

*Proof.*

□

#### **4.11 Exercise 11**

4.11.1 ()

*Proof.*

□

#### **4.12 Exercise 12**

4.12.1 ()

*Proof.*

□

#### **4.13 Exercise 13**

4.13.1 ()

*Proof.*

□

#### **4.14 Exercise 14**

4.14.1 ()

*Proof.*

□

#### **4.15 Exercise 15**

4.15.1 ()

*Proof.*

□

#### **4.16 Exercise 16**

4.16.1 ()

*Proof.*

□

#### **4.17 Exercise 17**

4.17.1 ()

*Proof.*

□

#### **4.18 Exercise 18**

4.18.1 ()

*Proof.*

□

## **4.19 Exercise 19**

**4.19.1**     $()$

*Proof.*

☐

## **4.20 Exercise 20**

**4.20.1**     $()$

*Proof.*

☐

## **4.21 Exercise 21**

**4.21.1**     $()$

*Proof.*

☐

## **4.22 Exercise 22**

**4.22.1**     $()$

*Proof.*

☐

## **4.23 Exercise 23**

**4.23.1**     $()$

*Proof.*

☐

## **4.24 Exercise 24**

**4.24.1**     $()$

*Proof.*

☐

## **4.25 Exercise 25**

**4.25.1**     $()$

*Proof.*

☐

## **4.26 Exercise 26**

**4.26.1**     $()$

*Proof.*

☐

## 4.27 Exercise 27

4.27.1 ()

*Proof.*



## 4.28 Exercise 28

4.28.1 ()

*Proof.*



## 4.29 Exercise 29

4.29.1 ()

*Proof.*



## 4.30 Exercise 30

4.30.1 ()

*Proof.*



## 4.31 Exercise 31

4.31.1 ()

*Proof.*



# 5 Exercise Set 10.5

## 5.1 Exercise 1

5.1.1 ()

*Proof.*



## 5.2 Exercise 2

5.2.1 ()

*Proof.*



## 5.3 Exercise 3

5.3.1 ()

*Proof.*



## 5.4 Exercise 4

5.4.1 ()

*Proof.*



## 5.5 Exercise 5

5.5.1 ()

*Proof.*



## 5.6 Exercise 6

5.6.1 ()

*Proof.*



## 5.7 Exercise 7

5.7.1 ()

*Proof.*



## 5.8 Exercise 8

5.8.1 ()

*Proof.*



## 5.9 Exercise 9

5.9.1 ()

*Proof.*



## 5.10 Exercise 10

5.10.1 ()

*Proof.*



## 5.11 Exercise 11

5.11.1 ()

*Proof.*



## **5.12 Exercise 12**

**5.12.1**    ()

*Proof.*



## **5.13 Exercise 13**

**5.13.1**    ()

*Proof.*



## **5.14 Exercise 14**

**5.14.1**    ()

*Proof.*



## **5.15 Exercise 15**

**5.15.1**    ()

*Proof.*



## **5.16 Exercise 16**

**5.16.1**    ()

*Proof.*



## **5.17 Exercise 17**

**5.17.1**    ()

*Proof.*



## **5.18 Exercise 18**

**5.18.1**    ()

*Proof.*



## **5.19 Exercise 19**

**5.19.1**    ()

*Proof.*





## 5.20 Exercise 20

5.20.1 ()

*Proof.*



## 5.21 Exercise 21

5.21.1 ()

*Proof.*



## 5.22 Exercise 22

5.22.1 ()

*Proof.*



## 5.23 Exercise 23

5.23.1 ()

*Proof.*



## 5.24 Exercise 24

5.24.1 ()

*Proof.*



## 5.25 Exercise 25

5.25.1 ()

*Proof.*



## 6 Exercise Set 10.6

### 6.1 Exercise 1

6.1.1 ()

*Proof.*



### 6.2 Exercise 2

6.2.1 ()

*Proof.*



### 6.3 Exercise 3

6.3.1 ()

*Proof.*



### 6.4 Exercise 4

6.4.1 ()

*Proof.*



### 6.5 Exercise 5

6.5.1 ()

*Proof.*



### 6.6 Exercise 6

6.6.1 ()

*Proof.*



### 6.7 Exercise 7

6.7.1 ()

*Proof.*



### 6.8 Exercise 8

6.8.1 ()

*Proof.*



### 6.9 Exercise 9

6.9.1 ()

*Proof.*



### 6.10 Exercise 10

6.10.1 ()

*Proof.*



## **6.11 Exercise 11**

**6.11.1** ()

*Proof.*



## **6.12 Exercise 12**

**6.12.1** ()

*Proof.*



## **6.13 Exercise 13**

**6.13.1** ()

*Proof.*



## **6.14 Exercise 14**

**6.14.1** ()

*Proof.*



## **6.15 Exercise 15**

**6.15.1** ()

*Proof.*



## **6.16 Exercise 16**

**6.16.1** ()

*Proof.*



## **6.17 Exercise 17**

**6.17.1** ()

*Proof.*



## **6.18 Exercise 18**

**6.18.1** ()

*Proof.*



## **6.19 Exercise 19**

**6.19.1**     $()$

*Proof.*



## **6.20 Exercise 20**

**6.20.1**     $()$

*Proof.*



## **6.21 Exercise 21**

**6.21.1**     $()$

*Proof.*



## **6.22 Exercise 22**

**6.22.1**     $()$

*Proof.*



## **6.23 Exercise 23**

**6.23.1**     $()$

*Proof.*



## **6.24 Exercise 24**

**6.24.1**     $()$

*Proof.*



## **6.25 Exercise 25**

**6.25.1**     $()$

*Proof.*



## **6.26 Exercise 26**

**6.26.1**     $()$

*Proof.*



## **6.27   Exercise 27**

**6.27.1**    $()$

*Proof.*



## **6.28   Exercise 28**

**6.28.1**    $()$

*Proof.*



## **6.29   Exercise 29**

**6.29.1**    $()$

*Proof.*



## **6.30   Exercise 30**

**6.30.1**    $()$

*Proof.*



## **6.31   Exercise 31**

**6.31.1**    $()$

*Proof.*

