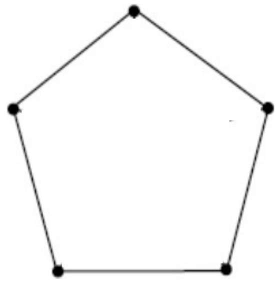
DAT157 Obligatorisk oppgave 2
(Algoritmer 1)

145157 Pål V. Gjerde

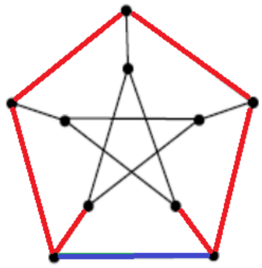
Problem 1

This can be shown fairly easily through a proof by contradiction, by assuming that the graph is 3-colourable and showing that this doesn't lead to a valid 3-colouring.



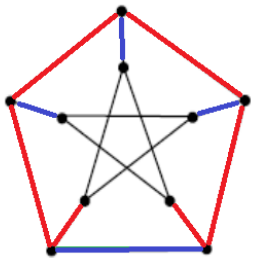
First, consider only the outside edges of the Peterson graph, depicted to the left. It is easy to see that regardless of colouring degree, it is impossible for any colour to appear more than twice in this cycle. With the assumption that the full graph is 3-colourable, that means this cycle also must be, which leads to the conclusion that two of the colours will cover two edges each, and the final colour will cover one edge.

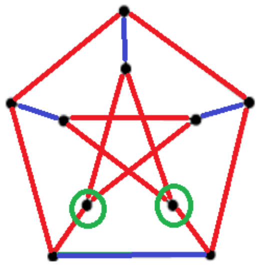
The strategy from this point will be to choose the colour that only covers one of these edges, and colour those edges blue. Any time an edge gets coloured blue, all edges that as a result of that cannot be blue will be coloured red.



In the first step, the bottommost edge is coloured blue. Due to the graph's symmetry, it does not matter which of the outer edges one starts with; this one was chosen arbitrarily. All other edges connected to those two nodes are also coloured red, signifying they cannot be blue. The top two edges are also coloured red, due to the fact that we chose blue as the colour that only covers one outside edge.

At this point, take note of the fact that each node has degree 3. This means that in a 3-colourable graph, each node must have an edge with each colour. This provides the necessary colouring for three more edges at this point, as seen here to the right.





In the final step, all edges that now cannot be blue are again coloured red. This causes all of the edges in the innermost portion of the graph to be coloured red. This also leaves two nodes, here circled in green, without any blue edges connected to them and no edges left that can be coloured blue. Thus, it is impossible to colour the graph using three colours.

Problem 2

a)
The distance from 1 to 5 directly is 11. The distance of going from 1 to 5 through 2 or 3 is also 11, and the distance of going through 4 or 6 is 13. Thus the triangle inequality holds.

b)
First, the shortest tour with 2 nodes is chosen, which is 3-6-3.
Then, the node with the shortest edge between it and nodes already added to the tour is chosen, adding it just after said node in the tour. This is repeated until all nodes are in the tour.
In this case:
3-6-3
3-1-6-3
3-1-2-6-3
3-1-2-4-6-3
3-1-2-4-5-6-3
This tour has length 26.

c)
The graph has three MSTs:
1. {(1,2), (2,4), (4,5), (1,3), (3,6}
2. {(1,2), (1,3), (3,5), (3,6), (4,5)}
3. {(1,2), (1,3), (3,6), (4,5), (5,6)}

Considering 1 to be the root of the tree, the double tree algorithm finds the nodes in this order for each of the MSTs of the graph. To the left is the order the nodes are visited; to the right, the same order but with repeat nodes (except the end node) removed.

1. 1-2-4-5-4-2-1-3-6-3-1	1-2-4-5-3-6-1	Total distance: 28
2. 1-2-1-3-5-4-5-3-6-3-1	1-2-3-5-4-6-1	Total distance: 38
3. 1-2-1-3-6-5-4-5-6-3-1	1-2-3-6-5-4-1	Total distance: 34

d)
The MSTs are the same as in **c)**. The first and third MST only have two nodes with odd degree each, which means there's only one possible matching for each – for the first, this is {(4,6)}, and for the last it's {(2,4)}. For the second tree, there are 4 nodes with odd degree. This leaves few enough possible matchings that finding the minimum one was fairly simple: {(2,4), (3,6)}. After adding these connections to the MSTs, the first and third trees actually contain the same tour. The second tree contains a different one.

1. 1-2-4-5-6-3-1	1-2-4-5-6-3-1	Total distance: 26
2. 1-2-4-5-3-6-3-1	1-2-4-5-3-6-1	Total distance: 28

Problem 3

a)

The randomized algorithm makes no effort to reach a correct conclusion, it just sets each variable randomly. Doing this once, I ended up with the following values:

0 1 1 0 0

b)

In this algorithm, the probability for any one clause to be true is calculated based on two possible assumptions for each variable, and the concluded truth value of the previous variables, with all variables succeeding the current being true with 1/2 probability; whichever possibility has the largest sum wins. In this table, this is done from left to right, starting with x_1 and going through to x_5. Note that in each column, only those clauses where the values differ are calculated; clauses that are true based on previous values and clauses that don't include the current variable at all are excluded. The chosen value for each variable is bolded.

	X_1: T	**X_1: F**	**X_2: T**	X_2: F	X_3: T	**X_3: F**	X_4: T	**X_4: F**	**X_5: T**	X_5: F
C_1	2.25	3								
C_2	2	1			0	2				
C_3	1	0.75					1	0.5	0	1
C_4	1.5	3								
C_5			4	2						
C_6			1	2					2	0
C_7					2	1	0	2		
C_8							2.25	3		
C_9									5	2.5
Sum	6.75	7.75	5	4	2	3	3.25	5.5	7	3.5

Problem 4

a)

A vertex cover is a subset of a graph's vertices (nodes) such that each edge is connected to at least one node in said subset. A minimum vertex cover is such a cover that contains as few nodes as possible. Another way to put this is that we want to minimise the number of nodes in the cover, subject to the restriction that each edge has to have a node in the cover. In this case, the graph has the following nodes and edges:

Nodes: {1, 2, 3, 4, 5}
Edges: {(1, 2), (1, 3), (1, 5), (2, 3), (2, 5), (4, 5)}

Note that the graph is not directed; that is, the edge from a to b is also an edge from b to a.

Since the edges don't have specific weights, they will all be considered to have the weight 1.

The integer linear programming problem version of this is as follows.

Minimize $x_1+x_2+x_3+x_4+x_5$
Subject to:

$$
\begin{array}{llllll}
x_i \in \{0, 1\} & & & & & \\
x_1 & + x_2 & & & & \geq 1 \\
x_1 & & + x_3 & & & \geq 1 \\
x_1 & & & & + x_5 & \geq 1 \\
 & x_2 & + x_3 & & & \geq 1 \\
 & x_2 & & & + x_5 & \geq 1 \\
 & & & x_4 & + x_5 & \geq 1
\end{array}
$$

b)

The linear programming relaxation is very simple – keep everything, except the first restriction, which is now simply that $x_i \geq 0$.

Minimise $x_1+x_2+x_3+x_4+x_5$
Subject to:

$$
\begin{array}{llllll}
x_i \geq 0 & & & & & \\
x_1 & + x_2 & & & & \geq 1 \\
x_1 & & + x_3 & & & \geq 1 \\
x_1 & & & & + x_5 & \geq 1 \\
 & x_2 & + x_3 & & & \geq 1 \\
 & x_2 & & & + x_5 & \geq 1 \\
 & & & x_4 & + x_5 & \geq 1
\end{array}
$$

c)

The above is a minimisation problem; the dual problem is a corresponding maximisation problem. In this case, it's not difficult to transform the problem:

Maximise $y_1+y_2+y_3+y_4+y_5+y_6$
Subject to:

$$
\begin{array}{lllllll}
y_i \geq 0 & & & & & & \\
y_1 & + y_2 & + y_3 & & & & \leq 1 \\
y_1 & & & + y_4 & + y_5 & & \leq 1 \\
 & y_2 & & + y_4 & & & \leq 1 \\
 & & & & & y_6 & \leq 1 \\
 & & y_3 & & + y_5 & + y_6 & \leq 1
\end{array}
$$