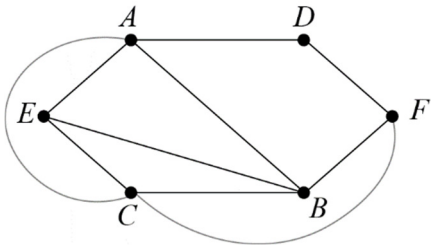
DECISION MATHS UNIT TEST 6 MARK SCHEME

Q	Scheme	Marks	AOs	Pearson Progression Step and Progress Descriptor
1a	For example, *ADFBCEA*, *ADFCBEA, ADFBECA,* or *ADFCEBA*	**B1**	1.1b	7th Be able to determine if a graph contains a Hamiltonian cycle
		(1)		
1b	For example, using *ADFBCEA,* *AB(I)* *AC(O)* *BE(I)* *CF(O)*  	**M1** **A2**	1.1b 1.1b	6th Be able to apply the planarity algorithm for planar graphs
		(3)		
1c	States that one named arc (either *ED* or *CD*) crosses at least one named arc	**B1**	2.4	6th Be able to apply the planarity algorithm for planar graphs
		(1)		
				(5 marks)

Notes

1a Must start and finish with *A*.

1b **M1** for Hamiltonian cycle drawn; **A1** for each non-intersecting arc drawn.

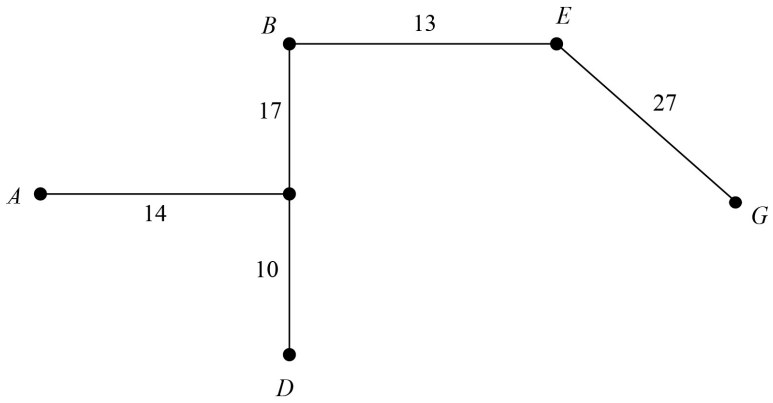
DECISION MATHS UNIT TEST 6 MARK SCHEME

<table>
<tr><th>Q</th><th>Scheme</th><th>Marks</th><th>AOs</th><th>Pearson Progression Step and Progress Descriptor</th></tr>
<tr><td rowspan="2">2a</td><td>
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr>
<tr><td>A</td><td>-</td><td>20</td><td>40</td><td>15</td><td>15</td></tr>
<tr><td>B</td><td>20</td><td>-</td><td>20</td><td>5</td><td>21</td></tr>
<tr><td>C</td><td>40</td><td>20</td><td>-</td><td>25</td><td>41</td></tr>
<tr><td>D</td><td>15</td><td>5</td><td>25</td><td>-</td><td>16</td></tr>
<tr><td>E</td><td>15</td><td>21</td><td>41</td><td>16</td><td>-</td></tr>
</table>
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr>
<tr><td>A</td><td>A</td><td>D</td><td>D</td><td>D</td><td>E</td></tr>
<tr><td>B</td><td>D</td><td>B</td><td>C</td><td>D</td><td>D</td></tr>
<tr><td>C</td><td>D</td><td>B</td><td>C</td><td>B</td><td>D</td></tr>
<tr><td>D</td><td>A</td><td>B</td><td>B</td><td>D</td><td>E</td></tr>
<tr><td>E</td><td>A</td><td>D</td><td>D</td><td>D</td><td>E</td></tr>
</table>
</td><td>M1
A1
M1
A1</td><td>1.1b</td><td>7th
Be able to find all shortest paths between all pairs of vertices using Floyd's algorithm</td></tr>
<tr><td></td><td>(4)</td><td></td><td></td></tr>
<tr><td rowspan="2">2b</td><td>Using row 3, column 1
Shortest distance from C to A is 40
From route matrix CA goes via D, so CDA
CD goes via B, from route matrix
So, CBDA</td><td>M1
A1
M1
A1</td><td>2.4</td><td>7th
Be able to find all shortest paths between all pairs of vertices using Floyd's algorithm</td></tr>
<tr><td></td><td>(4)</td><td></td><td></td></tr>
<tr><td colspan="5">(8 marks)</td></tr>
<tr><td colspan="5">Notes
2a M1 first row distances
A1 All distances
M1 row C correct routes
A1 All routes
Condone dashes top left to bottom right diagonal</td></tr>
</table>

<table>
<tr><th>Q</th><th>Scheme</th><th>Marks</th><th>AOs</th><th>Pearson Progression Step and Progress Descriptor</th></tr>
<tr><td rowspan="6">3a</td><td>
<table>
<tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>A</td><td>-</td><td>2</td><td>6</td><td>4</td></tr>
<tr><td>B</td><td>2</td><td>-</td><td>3</td><td>1</td></tr>
<tr><td>C</td><td>6</td><td>3</td><td>-</td><td>1</td></tr>
<tr><td>D</td><td>4</td><td>1</td><td>1</td><td>-</td></tr>
</table>
</td><td>M1
A1</td><td>1.1b</td><td rowspan="6">7th
Be able to find all shortest paths between all pairs of vertices using Floyd's algorithm</td></tr>
<tr><td>
<table>
<tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>A</td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>B</td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>C</td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>D</td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
</table>
</td><td>A1ft</td><td></td></tr>
<tr><td>
<table>
<tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>A</td><td>-</td><td>2</td><td>6</td><td>4</td></tr>
<tr><td>B</td><td>2</td><td>-</td><td>3</td><td>1</td></tr>
<tr><td>C</td><td>6</td><td>3</td><td>-</td><td>1</td></tr>
<tr><td>D</td><td>4</td><td>1</td><td>1</td><td>-</td></tr>
</table>
</td><td>M1
A1</td><td></td></tr>
<tr><td>
<table>
<tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>A</td><td>-</td><td>2</td><td>5</td><td>3</td></tr>
<tr><td>B</td><td>2</td><td>-</td><td>3</td><td>1</td></tr>
<tr><td>C</td><td>5</td><td>3</td><td>-</td><td>1</td></tr>
<tr><td>D</td><td>3</td><td>1</td><td>1</td><td>-</td></tr>
</table>
</td><td>A1</td><td></td></tr>
<tr><td>
<table>
<tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>A</td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>B</td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>C</td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>D</td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>A</td><td>A</td><td>B</td><td>B</td><td>B</td></tr>
<tr><td>B</td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>C</td><td>B</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>D</td><td>B</td><td>B</td><td>C</td><td>D</td></tr>
</table>
</td><td>A1
A1</td><td></td></tr>
</table>

		(9)		
3b	*A* to *C* via *D* (row 1, col 3) so *ADC* *A* to *D* is via *B* (row1 col 4) so *ABDC* (row 1, col 3) from distance matrix Distance = 4	**M1** **A1** **A1**	2.4	7th Be able to find all shortest paths between all pairs of vertices using Floyd's algorithm
		(3)		
				(12 marks)

	Notes
3a	First **M1:** All 32 cells completed, with dashes on diagonal. Condone 2 errors per table.
	First **A1:** All distances and routes correct.
	Second **M1:** Allow one error only, so at least one value reduced correctly
	Fifth **A1:** *D* in row 1 col 3
	Sixth **A1:** All correct

<table>
<tr><td rowspan="4">4a</td><td> 
</td><td>M1</td><td rowspan="4">1.1b</td><td rowspan="4">7th
Know how to identify the best lower and upper bounds</td></tr>
<tr><td>Length = 81 km</td><td>A1</td></tr>
<tr><td>So lower bound 81 + 15 + 18 = 114 km</td><td>M1
A1</td></tr>
<tr><td>Best lower bound is 122 km</td><td>B1</td></tr>
<tr><td></td><td></td><td>(5)</td><td></td><td></td></tr>
<tr><td rowspan="2">4b</td><td><table>
<tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td></tr>
<tr><td>A</td><td>-</td><td>20</td><td>14</td><td>24</td><td>33</td><td>29</td><td>50</td></tr>
<tr><td>B</td><td>20</td><td>-</td><td>17</td><td>27</td><td>13</td><td>31</td><td>40</td></tr>
<tr><td>C</td><td>14</td><td>17</td><td>-</td><td>10</td><td>19</td><td>15</td><td>36</td></tr>
<tr><td>D</td><td>24</td><td>27</td><td>10</td><td>-</td><td>29</td><td>25</td><td>30</td></tr>
<tr><td>E</td><td>33</td><td>13</td><td>19</td><td>29</td><td>-</td><td>18</td><td>27</td></tr>
<tr><td>F</td><td>29</td><td>31</td><td>15</td><td>25</td><td>18</td><td>-</td><td>21</td></tr>
<tr><td>G</td><td>50</td><td>40</td><td>36</td><td>30</td><td>27</td><td>21</td><td>-</td></tr>
</table></td><td>B1
B1
B1</td><td rowspan="2">1.1b</td><td rowspan="2">5th
Understand the travelling salesman problem</td></tr>
<tr><td></td><td>(3)</td></tr>
<tr><td rowspan="3">4c</td><td>FCD(C)ABEGF</td><td>M1
A1</td><td rowspan="2">1.1b</td><td rowspan="2">5th
Use the nearest neighbour algorithm to find an upper bound</td></tr>
<tr><td>Length 130 km</td><td>A1</td></tr>
<tr><td></td><td>(3)</td><td></td><td></td></tr>
<tr><td colspan="5">(11 marks)</td></tr>
<tr><td colspan="5">Notes
4b B1 DF correct. B1 AC correct. B1 all others correct.
4c M1Tour, all vertices visited.</td></tr>
</table>

DECISION MATHS UNIT TEST 6 MARK SCHEME

Q	Scheme	Marks	AOs	Pearson Progression Step and Progress Descriptor
5a	Two shortest arcs from depot A and E 11 + 12 = 23	**M1** **A1**	3.1b	8th Solve the travelling salesman problem and interpret the solution in context
	16 + 23 = 39 km	**A1**		
		(3)		
5b	*PAEDBCFGP*	**M1** **A1**	1.1b	8th Solve the travelling salesman problem and interpret the solution in context
	Upper bound 47 km	**A1**		
		(3)		
				(6 marks)

Notes

5b **M1** correctly uses spanning tree (*P*)*AEDBC*…..

TOTAL MARK: 42