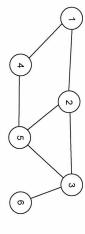
7 (c) Figure 5 from page 17 is repeated here so that you can answer Question 7(c) without having to turn back in the question booklet.

Figure 5 (Repeated)



A simple method of determining the shortest path through a network from one router to another is to perform a breadth first search of the graph representation of the network.

The algorithm in Figure 6 can be used to perform a breadth first search of a graph. It makes use of two subroutines, PutVertexIntoQueue and GetVertexFromQueue, which are explained below the algorithm.

Figure 6

```
Procedure ShortestRoute(S, D)
                                                                                                                                                                                                                                                                                                          While Queue is Not Empty And Found = False V = GetVertexFromQueue
                                                                                                EndWhile
                                                                                                                                                                                                                                                                                                                                                            Found ← False
                                                                                                                                                                                                                                                                                                                                                                                  Discovered[S] ← True
                                                                                                                                                                                                                                                                                                                                                                                                        PutVertexIntoQueue(S)
                                                                          If Found = True Then
Repeat
                        Output D
                                                                                                                        EndFor
                                                     C \uparrow D
                                                                                                                                                                                                                                                                                       For each vertex U which is adjacent to V Do
                                                                                                                                              EndIf
                                                                                                                                                                                                                                                               If Discovered[U] = False And Found = False Then
                                                                                                                                                                    If U = D Then Found ← True
                                                                                                                                                                                                                   Discovered[U] ← True
                                                                                                                                                                                         Parent[U] ← V
                                                                                                                                                                                                                                       PutVertexIntoQueue(U)
                                                                                                                                                                                                                                                                                                                                      Do
```

PutVertexIntoQueue is a subroutine that adds a vertex to the rear of a queue.

7 (d)

Explain why it is useful to find the shortest path through the network

EndProcedure

EndIf

Until C = S

C ← Parent[C]
Output C

- GetVertexFromQueue is a subroutine that returns the name of the vertex at the
- front of the queue and removes it from the queue.



M/Jun15/COMP3

Complete the trace table below to show how the <code>Discovered</code> and <code>Parent</code> arrays, the variable <code>Found</code> and the queue contents are updated, together with what output is produced by the algorithm when it is called using <code>ShortestRoute(1, 6)</code>.

19

Before the algorithm is carried out, all cells in the Discovered array are set to the value False and the queue is empty.

The values of the variables $S,\,D,\,V,\,U$ and C have already been entered into the table for you.

The letter F has been used as an abbreviation for False. You should use T as an abbreviation for True.

[6 marks]

														1	\boxtimes	V.	,
														Q	X	t	,
						ω		4			N		Ъ		X	<	:
				0	5	2	5	1	5	ω	1	4	2		\times	_	
<u> </u>	N	ω	Q												\times	C	,
																Queue Front Rear	,
															hzj	-	
															μij	N.	,
															ıπl	u	,
															描	42	
															12]	υ	1
															H	σ	о 2 2 7 8
															\geq	-	1
															\boxtimes	N)
															\geq	u	_
															\bowtie	4	
															\boxtimes	U	1
															\times	σ	-
															X	F.ound	
															X	Found Output	

0

Turn over ▶

[1 mark]