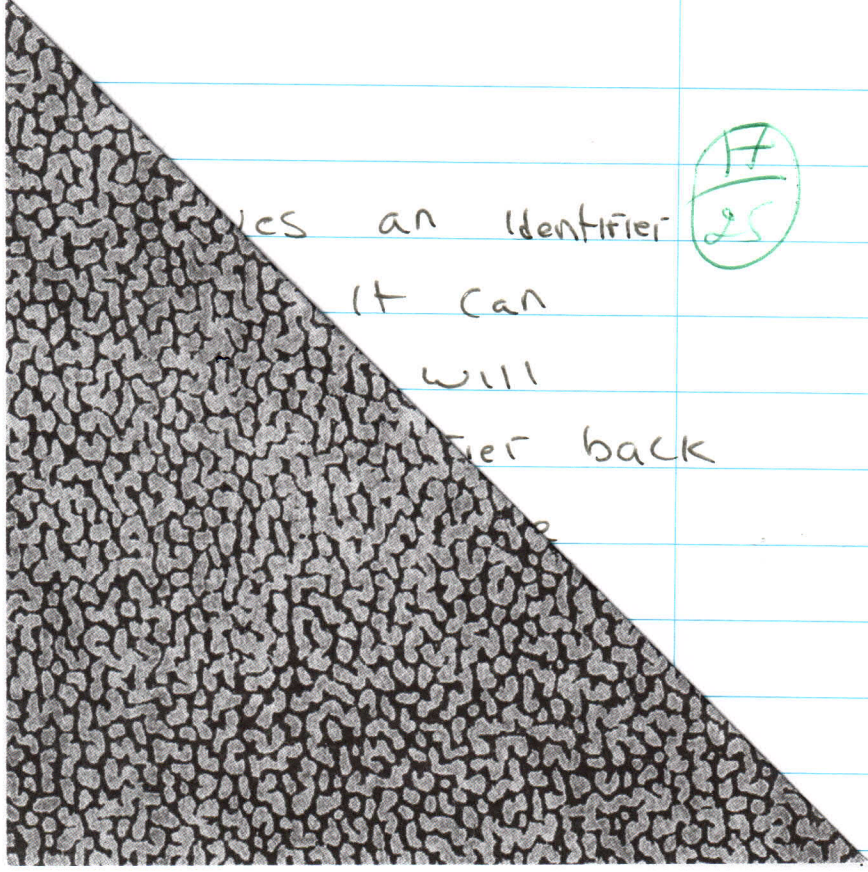
399

Coláiste na hOllscoile
Corcaigh
University College Cork



17/25

Uimhir Scrúdaithe
Examination Number | 9 | 1 | 7 | 1 | 6 | 3 | |

Module Code CS4616

Paper No. _____

Mír
Section _____

Calculator, Please state:	No. of Books submitted
Name Casio	2
Model Fx911ES	

Note: If there are different sections on this paper, a separate Answer Book MUST be used for each section.

Do na Scrúdaitheóirí amháin **For Examiner's use only**	
1	17/25
2	10/15
3	17/20
4	17/20
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
Iomlán **Total**	61/80

Q1

17/25

A Once a process receives an identifier larger than its own it can know with certainty it will never receive its own identifier back this is because there is some process in the network that will stop its token from progressing the entire way around the network

5/5

This simplifies or trans function to

Let v = our UID,

IF message ∈ M then

(Case v = M status := leader

Case v < M send := m

Q1B Me (Id, round, hop, bit)

States i

round $\in \mathbb{N}^+$, the current round for this process

send $\in M$, the message that will be sent to a neighbor

initial value of send = (UIDi, 0, N, FALSE)
(N: size of graph)

rand i

trans i

Send := null

IF message is (Id, round, hop, bit)

IF hop < N && Id = v

Send := (Id, round, hop+1, TRUE)

else IF hop = N && Id = v && bit = TRUE

go to next round →

round := round + 1

Send :=

States:

round $\in \mathbb{N}^+$, the current round for this proc

send $\in M$, the current message to be sent to the neighbouring process

has_next_round = False, is there another round

status $\in$ {leader, non leader, unknown}

the current status for process, inital unknow

CID $\in \{1, \ldots k\}$ the chosen random id using rand;

N the size of this ring network

Initial value of send is = (CID, 0, 1, False

M is of type (Id, round, hop, bit) where Id is this processes id, round is the current round, hop is the current number of hops this message has been through and bit signals whether this process has sent a message with conflicting UID

A process progresses rounds if it receives a message to it with hop of N and bit true with its UID.

If there is another round leader is chosen from processes in that round If there is not another round leader is chosen from this process

rand i

generate a random number between 1 and k

msgs i

send the value of send to i+1

trans i

IF Message of type M (part of our program)

Message is (ID, R, H, B)

12/12

Case ID = CID: ID is same as our ID

Case H < N: Send := (ID, R, H+1, T) (we didn't send this message)

has_next_round := TRUE

Case H = N We sent this message

Case B = TRUE It was detected as dupe

Round := Round +1 move to next round

rand() generate new random num

Send = (CID, Round, hop, bit)

Send a new random number to try be a leader

CASE B = FALSE we got message back it was not dupe, it must be highest

IF has_next_round: there were duplicates! There will have to be another round

do_nothing (NOP)

else no duplicates detected

status := Leader

Case B = TRUE

has_next_Round = TRUE

Q.2

It does not apply to arbitary sorting algorithms ✓

Imagine a sort algorithm constructed as follows

Let A = Array to sort

* Operation Start A Move current item to start of the array A
* Operation End A Move current item to end of the array A
* MAX A, biggest item OF A → THIS WOULD BE SELECTION SORT, WHICH SORTS ALL ITEMS!

WHILE Ai ← Next item OF A

Case:

AI = MAX(A) : END A Ai

Otherwise : START A Ai → ARGUMENT LACKS CLARITY

10/15

After one Pass through A the Maximums will be Pushed to the end of the array and the mins will be at the start.

This will sort any combinations of 0 and 1 any other inputs where there is 2 or less (types of) items in the array but will not work for arbitrary arrays such as

(1, 2, 3) ⇒ (2, 1, 3)

?

Q3

A It does this in 3 stages, a random number is chosen by a process and sent to all its outgoing neighbours. Each process does this, ... if it is greater than the max of all other numbers it has received it declares itself the local winner and puts itself in the the independent set and communicates this to all its neighbours. The losers knowing who they are now remove themselves from the graph by communicating with all their neighbours that they have lost

It Yields an independent set because at each round there can be no two winners connected to each other added to the independent set

b

Iterative scheme

Start at an arbitrary node N.
and token I containing nodes in the MIS, initially {}
and token J containing nodes NOT in the MIS

current_node := N

while I ∪ J =/= Graph → MAY NEVER HAPPEN ...

Send a search message containing I
to all out nbrs OF current node
{a process responds with whether it is
{in I or not to the search requester

IF FOR all search responses:

All Processes not in I:
→ I = I ∪ {current_node}
→ J = J ∪ {out nbrs current node}

Otherwise
→ J = J ∪ {current node}

3/6

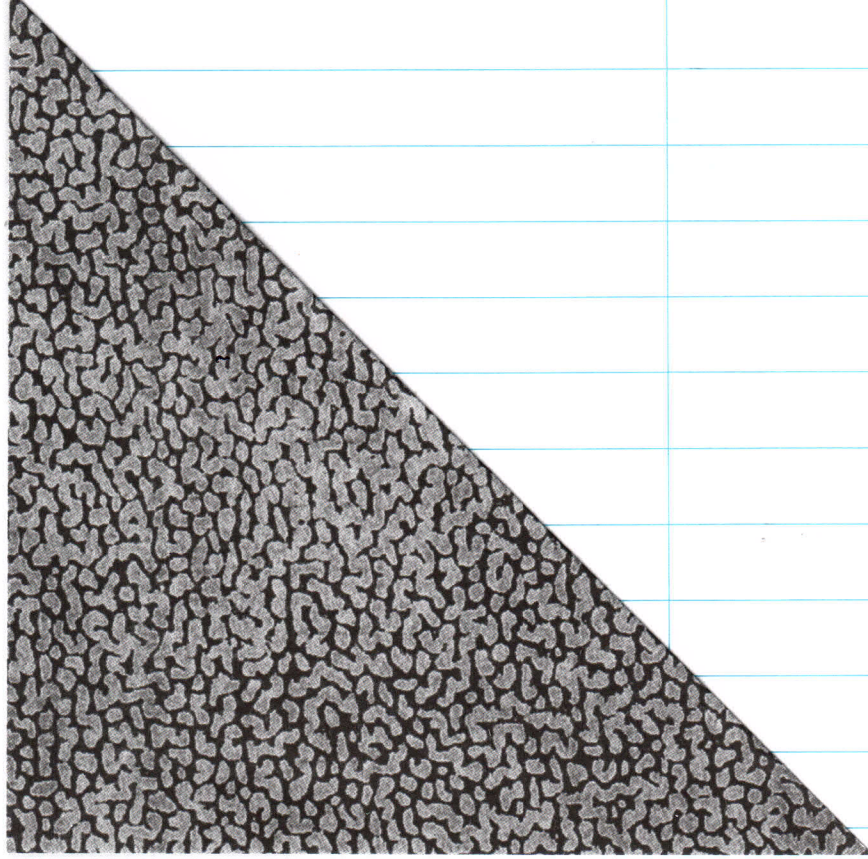
removed edges etc not described?

the tokens I and J are then passed along to the next node n in graph

This solution while inefficient works

399

Coláiste na hOllscoile
Corcaigh
University College Cork



Uimhir Scrúdaithe
Examination Number 9 1 7 1 6 3

Module Code CS4616

Paper No. _____

Mír
Section _____

Do na Scrúdaitheóirí amháin **For Examiner's use only**	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
Iomlán **Total**	

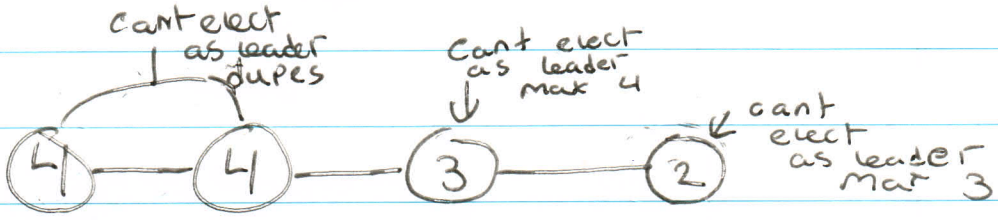
Calculator, Please state:	No. of Books submitted
Name Casio	2
Model Fx9160	

Note: If there are different sections on this paper, a separate Answer Book MUST be used for each section.

Q3

c

7/7

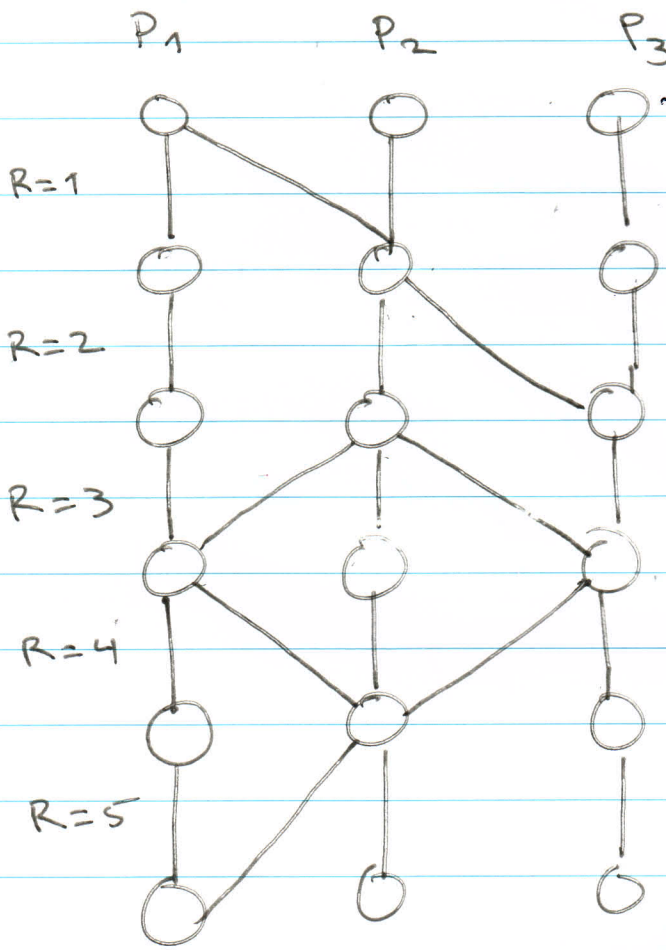

cant elect as leader dupes
cant elect as leader mar 4
cant elect as leader mar 3
4
4
3
2

✓

Q4 7/10

A 10 the outcome is 0 2/2

B



Graph: 3/3

Q4 C

Information Levels

0	0	0
0	0	0
0	0	1
0	0	1
0	1	1
1	1	1

Should be 2

9/10

PROB B

Q4 D

Decisions of Proc

Key	Proc 1	Proc 2	Proc 3	Reason
1	1	1	1	k = Max(1,1,1)
2	0	0	0	k > Max(1,1,1)
3	0	0	0	
4	0	0	0	
5	0	0	0	

3/5

W/O DISAGREEMENT WOULD YOU HAVE YOU MODE RECNEUR MATTERS?

PROB B = 0/5 ← they agree always as they have same INFO level