Name: _____ Student Number: _____
Please Print

Signature: _____

Instructor: J. Ellis

UNIVERSITY OF VICTORIA
DECEMBER EXAMINATIONS 1996
Computer Science 225 -- F01

Question	Value	Mark
1	50	
2	50	
3	50	
4	50	
5	50	
6	50	
7	50	
8	50	
9	50	
10	50	
Total	**500**	

Instructions:
ALL QUESTIONS MUST BE ANSWERED ON THE EXAMINATION PAPER IN THE SPACE PROVIDED.
This exam consists of 14 pages + cover.
This is a closed book exam. DURATION: 3 HOURS

Notation:
Throughout, = denotes the relational operator equality, <-- denotes the assignment operator.

Question 1 [50 marks]

Consider a Binary Search Tree that contains, for the sake of argument, words.

a) Indicate which of the following possible tree traversal orders will output the words in lexicographic order. (Assume that words are output in the traversal order.) Circle correct answer.

in-order post-order pre-order

b) Suppose we run Depth First Search on a Binary Search Tree, starting at the root. In which of the following possible orders are the nodes "visited"? Circle correct answer.

in-order post-order pre-order

c) Here is an informal definition of a traversal process suitable for the Binary Search Tree.

procedure traverse (v);

if (left child of v exists) **then** traverse (left child of v);

if (right child of v exists) **then** traverse (right child of v);

Where should we place an "output (word at node v)" statement, so that the words are output in lexicographic order? Assume we start the process by invoking the procedure on the root. Circle correct answer.

- before both **if** statements?
- in between the **if** statements?
- following both **if** statements?

Question 2 [50 marks]

Consider the Euclid's algorithm for finding the greatest common divisor of two positive integers, x and y. Assume that **mod** computes the remainder after integer division.

```
function gcd(x, y);

integer t;

while not (y mod x = 0) do
  t <-- y mod x; y <-- x; x <-- t
endwhile;

return x
```

a) Show that after two iterations, x is reduced by at least a factor of 2, by considering two cases: $x \geq y/2$ and $x \leq y/2$.

b) Now prove the best upper bound you can on the time complexity of the procedure.

c) Assume the following mathematical fact:

if x divides y (no remainder) then $\gcd(x,y) = x$, else $\gcd(x,y) = \gcd$ (remainder of y integer divided by x, x)

State a loop invariant assertion that is suitable for proving that the loop does what we want it to do.

d) Show that, if this assertion really is invariant, then x is indeed the correct answer at exit from the loop.
Note: You are NOT asked to show that it is invariant.

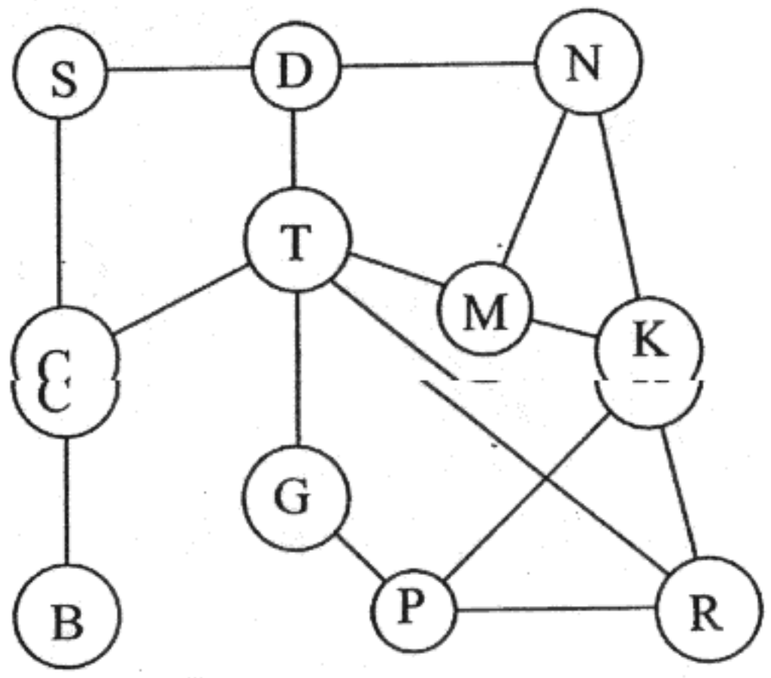
e) Prove that the procedure must terminate.

Question 3 [50 marks]

Consider the following algorithm which uses a FIFO queue, Q. to traverse an undirected graph.

```
procedure traverse (v);
enqueue (v, Q);
while not (Q is empty) do
  dequeue (Q, x);
  mark x as visited;
  for all unmarked neighbours y of x do
    enqueue (y, Q)
  endfor
endwhile
```

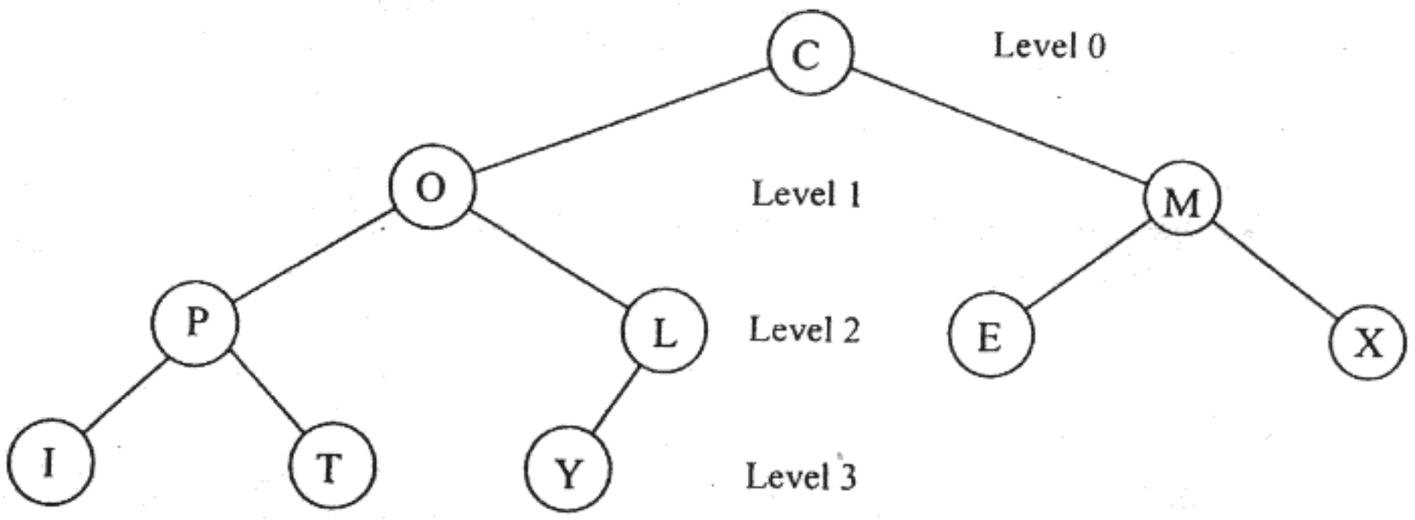
a) Number each node in the following graph in the order in which it is visited by this traversal procedure, assuming that the process starts at node **S**. Assume that neighbours of x are enqueued in lexicographic order.



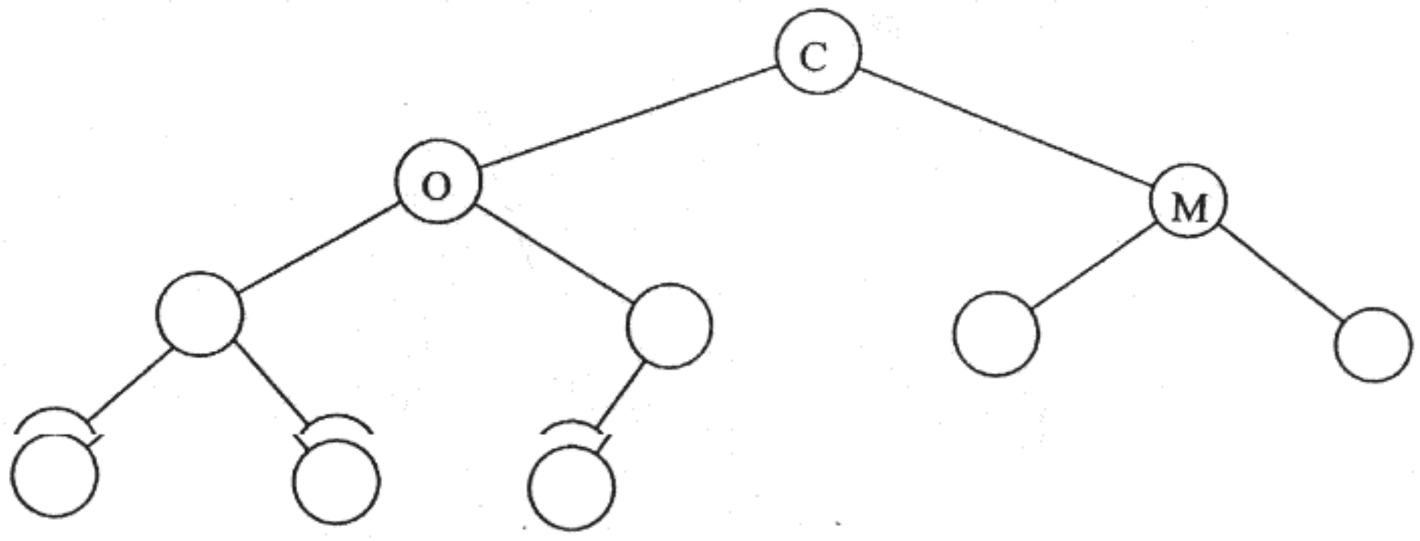
b) Assume that this traversal process does visit all nodes at distance d from the start node before it visits any node at distance $d+1$.
Show how to make simple additions to the procedure so that each node can be associated with its distance from the start node. Show where to insert statements by writing in the code above.

Question 4 [50 marks]

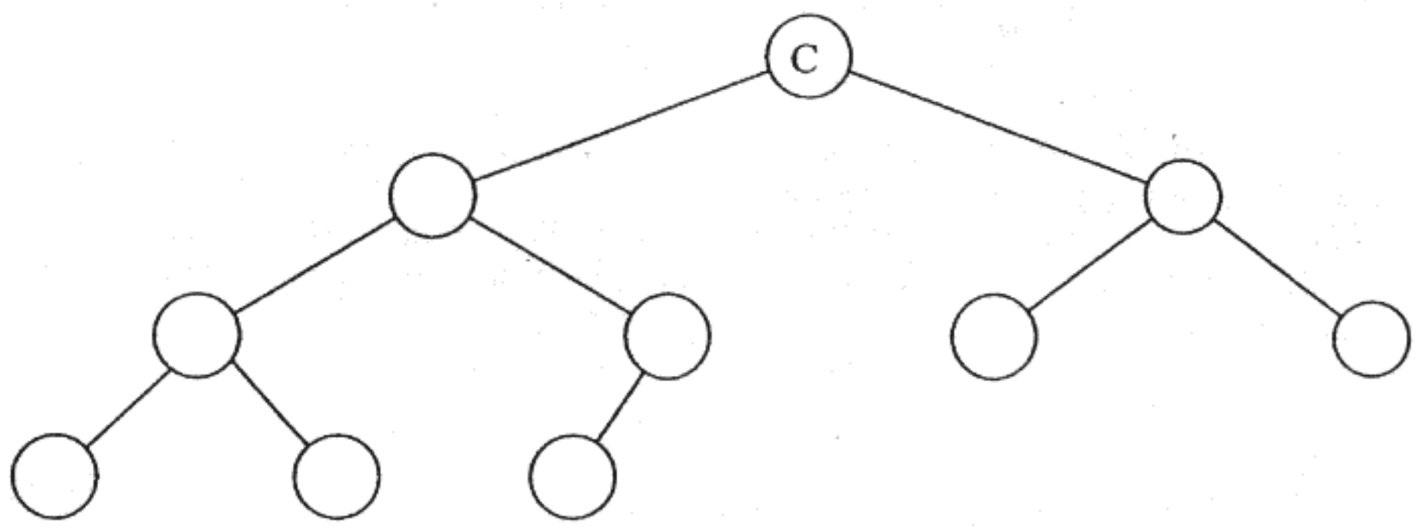
Remember the standard heap building process. That process begins by throwing the elements into the left complete binary tree at random. Then the elements are systematically rearranged so as to establish the heap property.
Now consider the example below. In the first figure, the elements, letters, have been thrown into the tree at random. Assume that element ordering is from "A" minimum to "Z" maximum and we want to order the heap with maximum on top.



a) Perform the first step of the rearrangement process, on the trees rooted at level 2.



b) Perform the second step of the rearrangement process, on the trees rooted at level 1.



c) State the relationship between the work done to completely build the heap and the height of each node.

d) Show by induction on the height of the tree (also called structural induction) that the sum of the heights of the nodes is at most n - 1, where n is the number of nodes.
Hint: Basis, height = 0, then show if it is true for 2 subtrees of a root, it is also true for the entire tree.

Question 5 [50 marks]

We have seen that the ADT set can be represented by a variety of data structures. The usefulness of a particular representation depends on which operations are to be done on the set. In the table on the next page each entry stands for a possible (operation, representation) pair. Enter the best upper bound for performing (worst case) the operation using the representation.

You need some assumptions in some cases, to make the question unambiguous.
Assume:

- The number of items entered into the hash table is not larger than the the number of slots in the table.
- The "vector" fits in one computer word.
- The union operation creates a new object, and does not affect the sets being operated on.
- A balanced binary tree has height $O(\log n)$, always.

TABLE FOR QUESTION 5

	Membership	Insert	Delete	Union
List (sorted, linked)				
List (unsorted, linked)				
List (sorted, array)				
Vector				
Balanced Binary Search Tree				
Hash Table (with chaining)				

Question 6 [50 marks]

a) Mention two essential properties of a good hash function.

b) Consider the following possible hash functions which should hash students in a class like CSC225 into a 100 slot hash table. For each say whether they are good or bad and why.

- Take the year of student's birthday, for example 1976, as a decimal number n and compute n mod 100 as the slot index.

- Take the day of the student's birthday, for example 23, as a decimal number n and compute n mod 100 as the slot index.

- Take the month of the student's birthday, for example 10, as a decimal number n, generate a random number r using the system's "rand" function or equivalent, and compute $n \times r$ mod 100 as the slot index.

- Take the 8 bit codes for the characters in the student's name, interpret each as an integer, add them all up to produce an integer n, and compute n mod 100 as the index slot.

c) A hash table with chaining has been filled with about 5 times as many items as there are slots in the table. About how many comparisons are expected to do a membership test on an element which is in fact not present? Assume a good hash function.

d) Is there any reason why an insert operation into a hash table with chaining should take longer when the table is sparsely occupied than when it is heavily loaded? Assume, we make it as fast as possible. Justify your response.

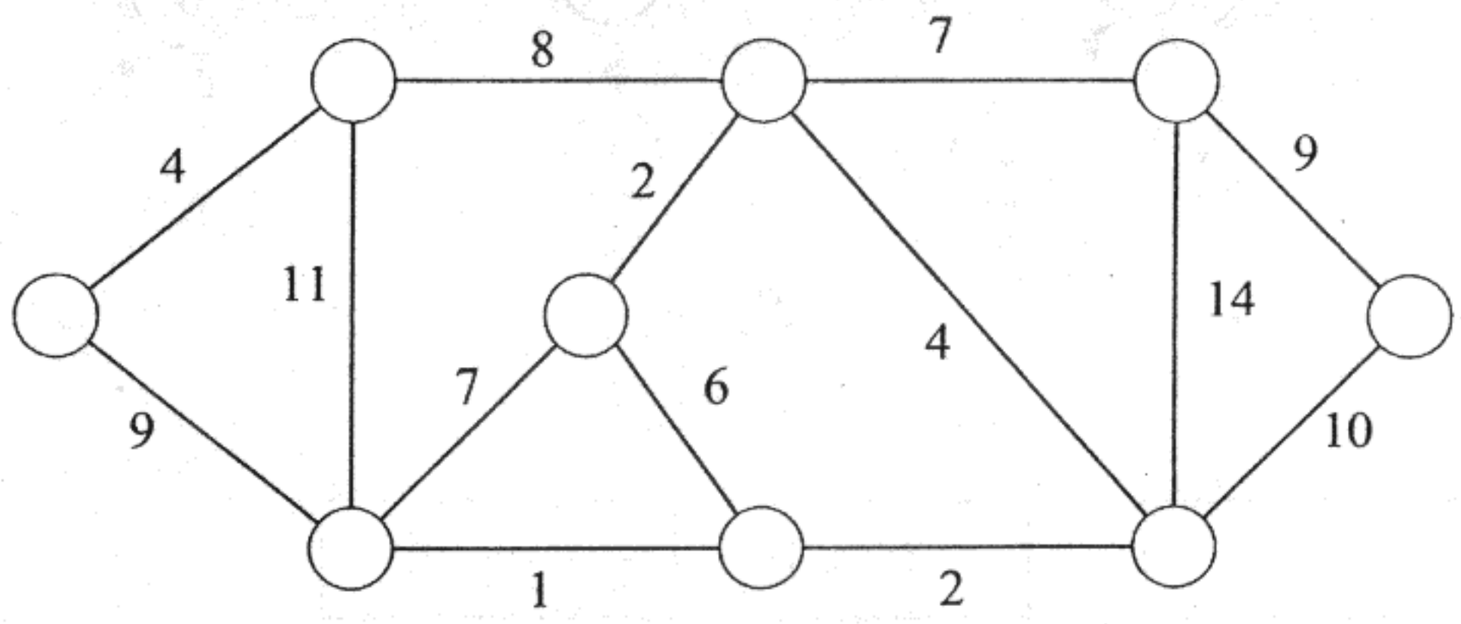
Question 7 [50 marks]

Here is a well known alternative to Kruskal's algorithm for computing a Minimum Weight Spanning Tree in the graph $G = \{V, E\}$. It is called Prim's algorithm.

T <-- {the least weight edge in E};

for k <-- 2 **to** $|V|$ - 1 **do**
  Condsider those edges with one end in T and one end not in T,
  choose the least weight such edge and add it to T;
endfor

a) Consider the graph below. Mark each edge that is in its Minimum Spanning Tree. Number the MST edges in the order in which they are chosen by Prim's algorithm.



b) It looks like we will need to maintain a set of edges connecting nodes in T to nodes not in T, and we want to be able to pull out the least weight edge quickly. What ADT should we use to represent this set?

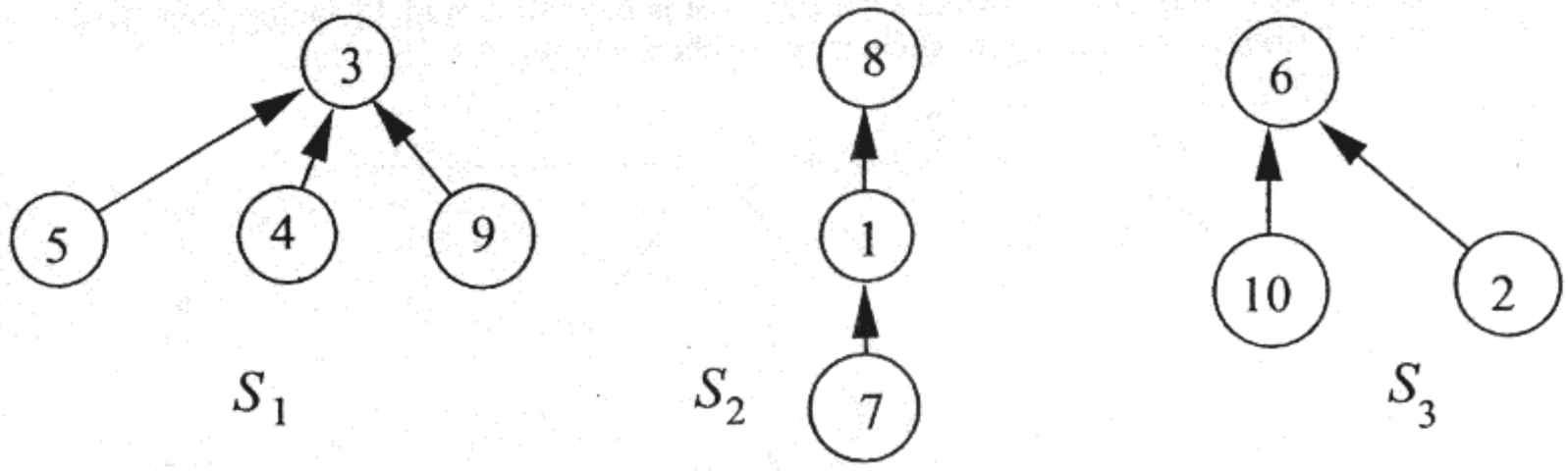
c) Outline in general terms what has to be done to update this ADT every time a least weight edge $e = \{u, v\}$ is removed, u in T and v not in T.

Question 8 [50 marks]

Remember the "Union/Find" problem, that is the ADT set with just two operations, union and find. (This is an imperative, not a question!) The sets that are being manipulated by these procedures are represented, in the abstract, by trees. But there is more than one way to represent a tree.

The trees representing the sets formed from the n elements could be represented by a one dimensional, n element, array where each element is the index of the parent of the current index, and a tree root is indicated by the element being its own index.

a) Using this idea, draw a 10 element array representing the following forest, (which represents 3 sets).



b) Which array element is changed to what value to effect a Union of S_1 and S_2?

c) Write a neat recursive function, Find (u), that returns the node identity of the node at the root of the tree containing u. Pseudo-code will do.
Hint: Something of the following form will do it.

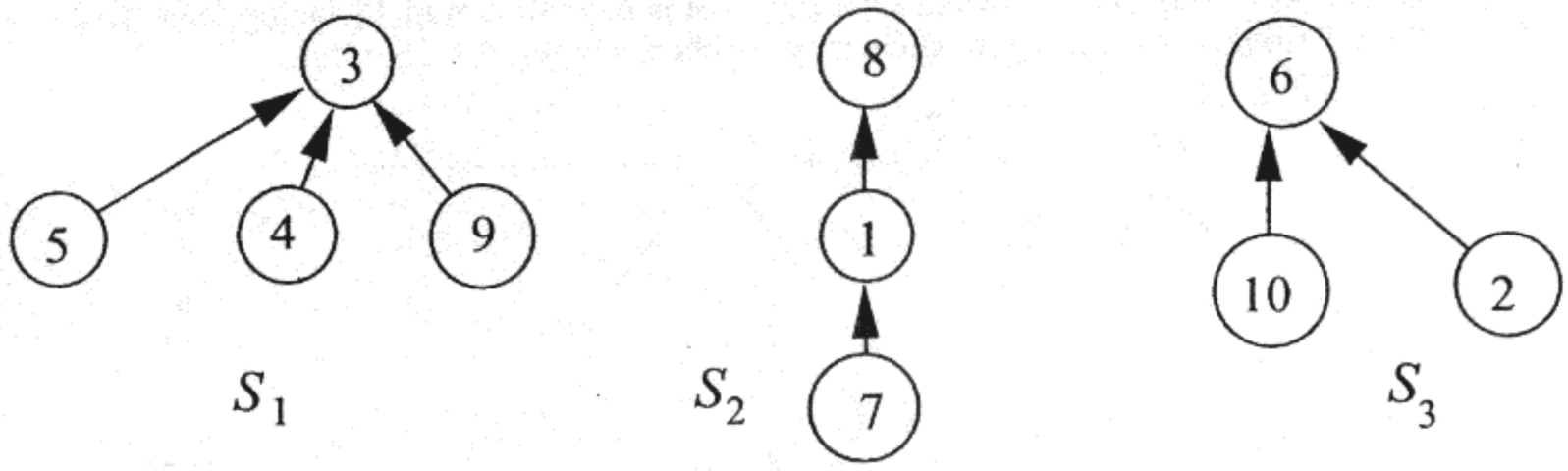
function Find (u);
if ???????? **then** return (????) **else** Find (????)

Question 8 [50 marks]

Remember the "Union/Find" problem, that is the ADT set with just two operations, union and find. (This is an imperative, not a question!) The sets that are being manipulated by these procedures are represented, in the abstract, by trees. But there is more than one way to represent a tree.

The trees representing the sets formed from the n elements could be represented by a one dimensional, n element, array where each element is the index of the parent of the current index, and a tree root is indicated by the element being its own index.

a) Using this idea, draw a 10 element array representing the following forest, (which represents 3 sets).



b) Which array element is changed to what value to effect a Union of S_1 and S_2?

c) Write a neat recursive function, Find (u), that returns the node identity of the node at the root of the tree containing u. Pseudo-code will do.
Hint: Something of the following form will do it.

function Find (u);
if ???????? **then** return (????) **else** Find (????)

Question 9 [50 marks]

Consider a directed, weighted, acyclic graph G. Assume all weights are positive, > 0. We wish to compute for all nodes u, the *maximum* weight path which starts at u. It can of course terminate on any other node.
Assume it is a good idea to start by doing a topological sort on G and renumbering the nodes in the order computed.

a) One can now see that for each node u, the maximum weight path can be computed in terms of the maximum weight path of which other nodes? State the relationship we need.

b) For what kinds of nodes is the maximum weight path of weight zero?

c) If the directed edges in the the topological numbering run from low to high numbers, outline the main loop of the process, including adequate detail in the body of the loop.

d) Assume that the graph is represented by an adjacency matrix and that the node numbering has been reordered to conform to the topological ordering. What is the time complexity of the algorithm, assuming that there are more edges than nodes? Justify your answer.

Question 10 [50 marks]

Consider the following outline of Dijkstra's shortest path algorithm.

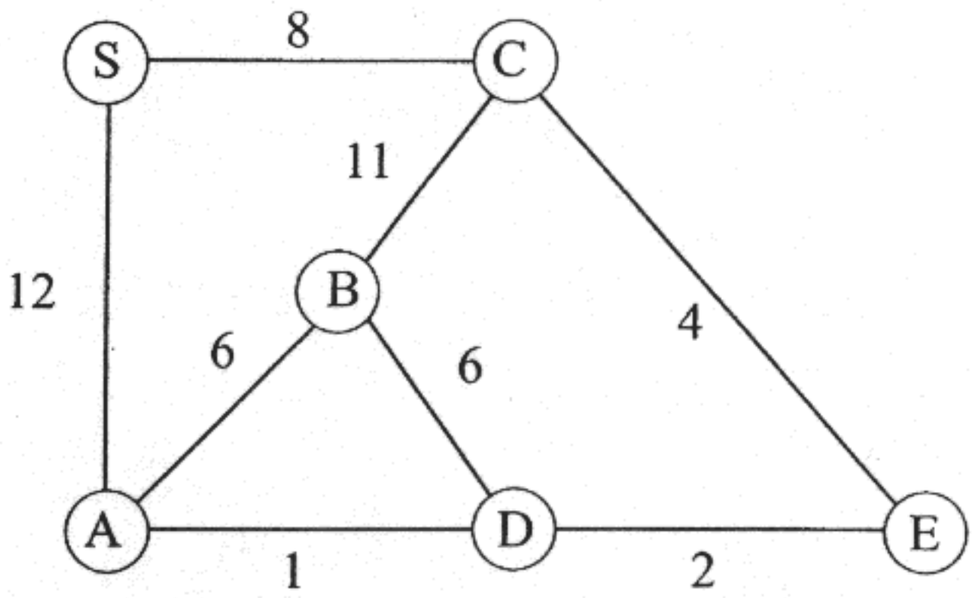
```
Initialise Settled and Unsettled sets;
Initialise distance matrix, dist[];

while not (Unsettled set is empty) do
  Let v be the unsettled vertex with minimum dist[v];
  Move v from Unsettled to Settled set;
  Update dist[u] for all u, Unsettled neighbours of v
endwhile
```

a) Specify in one line of pseudo-code exactly what the update process does for each unsettled node u that is adjacent to the new settled node v.

15 BONUS MARKS FOR THIS PART !!!

b) Describe a technique whereby the algorithm can be extended, so that at termination one could read out the actual shortest path (a node sequence, not just the length) from any node to the source node.
We need to leave "footprints" that for each node u tell us which was the preceding node in the shortest path to u.
So assume that for each node u we set up a location, say footprint[u], to record this preceding node.
At what point in the procedure should we update this "footprint"?

c) Consider the following weighted graph.
Fill in the columns of the following table, one column per repetion of the loop, where each column specifies the values of the distance matrix at each loop repetion.



round

node	1	2	3	4	5	6
S	0*					
A	11					
B	inf					
C	8					
D	inf					
E	inf					

END OF THE EXAM!!!