Name: _____

ID Number: _____

UNIVERSITY OF VICTORIA

EXAMINATIONS- DECEMBER 1998

CSC 225 F01

Instructor: Dr. W. Myrvold

Duration: 3 hours

TO BE ANSWERED ON THE PAPER.

Instructions:

Students **MUST** count the number of pages in this examination paper before beginning to write, and report any discrepancy immediately to the invigilator.

This question paper has 7 pages (the last page is blank in case you need extra space) plus the header page.

Use only space provided on exam for answering questions. Closed book. No aids permitted.

Question	Value	Mark
1	20	
2	20	
3	20	
4	20	
5	20	
Total	100	

1. [20] Circle true or false for each question and justify your answer. No marks will be given unless there is a correct justification.

(a) An algorithm for sorting n numbers which is $O(n)$ in the worst case is always faster than an algorithm which is $O(n^3)$ in the worst case.
True **False**

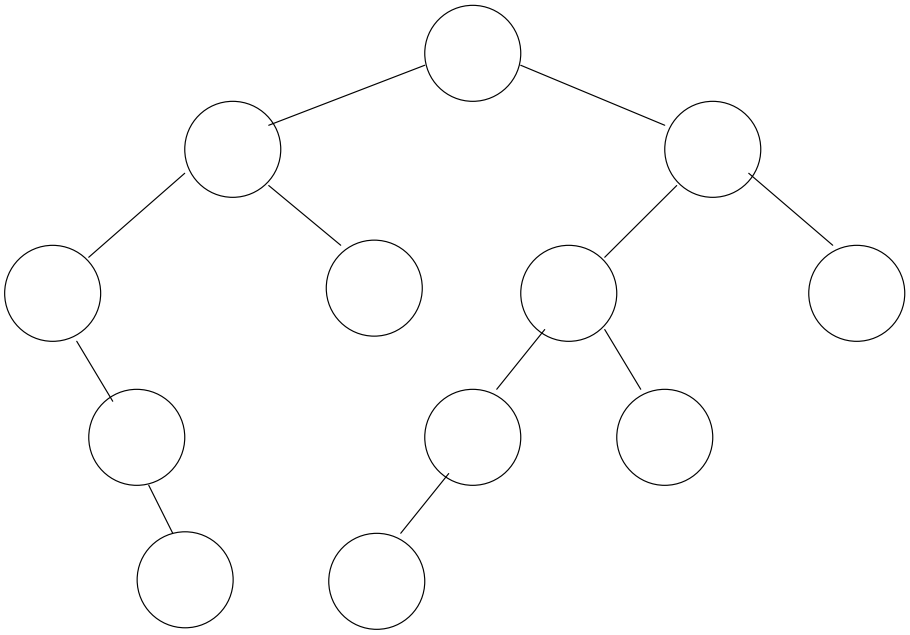
(b) Since it takes at least $n - 1$ key comparisons to find the maximum of n data items, it takes at least $2n - 2$ to find both the maximum and the minimum.
True **False**

(c) A heap can be built in $O(n)$ time.
True **False**

(d) Recursive algorithms can always be implemented without using recursion.
True **False**

2.(a) [10] Insert keys into the binary tree structure given so that a **postorder** traversal will give:

8	3	5	11	14	4	2	1	9	7	6	0



(b) [10] Suppose you have a tree with the adjacency list equal to:

0: 1 3 4 7
1: 0
2: 6
3: 0
4: 0 5 6
5: 4
6: 2 4
7: 0

Show how this tree could be represented in an array of size 8. Explain what you are doing.

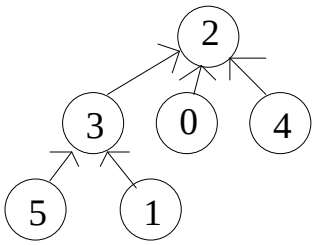
0	1	2	3	4	5	6	7

3. Suppose that we use a non-collapsing find and a union which always sets the root of the smaller tree to point to the root of the larger tree. Give the pseudocode for such a *union* and *find* which uses an array *parent* to store the data structure.

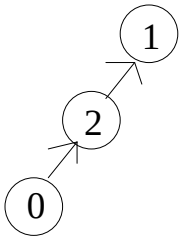
(a) [5] *int find(u, parent)*	(b) [5] *void union(u, v, parent)*

Which of the following can be created from by applying *find* from (a) and *union* from (b) starting with an empty data structure? To get credit, you must justify your answers.

(c) [5]



(d) [5]



4. The following algorithm was proposed for the MST problem:
   Repeat:
      Find a cycle C in G.
      Delete the maximum weight edge on C.
   Until n-1 edges remain.

(a) [15] Suppose that BFS is used as a tactic to find a cycle at each step. The BFS is terminated as soon as a cycle is found. How much time would such a BFS take in the worst case to find a cycle? Assume that the graph is stored using adjacency lists. Fully justify your answer.

(b) [5] Analyze the worst case time complexity of the MST algorithm given above as a function of both *n* (the number of vertices), and *m* (the number of edges).

5. Consider the following divide and conquer approach for reversing the order of the elements on a linked list.

   reverse_order(n, start, end)

   Input: n- the number of items in the list.
   start- a pointer to the start of the list.

   Output: start- points to the first cell of a new list having the keys in the reverse order as the original list.
   end- a pointer to the last cell on the list.

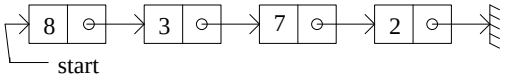
The divide and conquer strategy you must implement to accomplish this is as follows:

1. Divide the list into two sublists L1 containing the first $\left\lfloor \frac{n}{2} \right\rfloor$ items and L2 which contains the remaining items.
2. Reverse L1 and L2 recursively.
3. Connect the two reversed lists together to get a reversal of the original list.

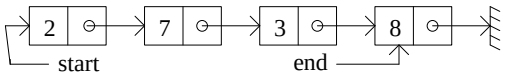
(a) [5] What is the worst case time complexity of this algorithm? Justify your answer by solving an appropriate recurrence.

(b) [15] Give detailed pseudo code (almost C code but without worrying about syntax) for the *reverse_order(n, start, end)* routine described on the previous page. To get full marks, your solution must implement the algorithm as specified.

If the initial list is:



On termination, the list is:



reverse_order(n, start, end)

Use this page if you need extra space. Clearly indicate the question you are answering.

The End.