Name: _____

ID Number: _____

UNIVERSITY OF VICTORIA

EXAMINATIONS- DECEMBER 1999

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 8 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	30	
2	15	
3	15	
4	10	
5	15	
6	15	
Total	100	

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

(a) [5] Suppose singly linked lists are used to implement a Queue. Then, adding to the Queue should be done at the front of the list, and deletions from the Queue should be taken from the end of the list.
True **False**

(b) [5] Sorting algorithms under the comparison model all require $\Omega(n \log_2 n)$ time and $\Omega(n)$ extra space (in addition to the space required to store the original data).
True **False**

(c) [5] Let a_0, a_1, a_2, and a_3 be integers where $a_3 > 0$. Then $p(n) = a_0 + a_1 n + a_2 n^2 + a_3 n^3$ satisfies $p(n) \leq (a_0 + a_1 + a_2 + a_3) n^3$ for $n \geq 1$.
True **False**

(d) [5] Given a graph G with distinct weights on its edges, the tree T created by the Dijkstra/Prim minimum spanning tree algorithm will be the same as the tree that the Dijkstra/Prim Shortest Paths algorithm creates.

True **False**

(e) [5] Any algorithm under the comparison model requires at least 7 key comparisons in the worst case to sort 5 data items.

True **False**

(f) [5] 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**

2. Consider the following program fragment:

```
answer=0;
x= 8;
for (i=1; i < k; i++)
{
   answer= answer +  i + x;
   x = x * 2;
}
```

Your goal is to determine a closed formula for the value of **answer** (as a function of *k*) when the loop terminates.

(a) [4] Set up an appropriate loop invariant. It should use closed formulas for **answer** and **x**.

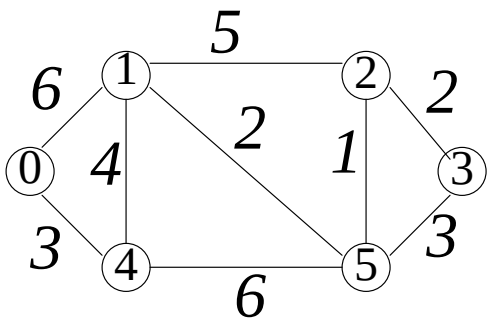
(b) [8] Prove by induction that your loop invariant is correct.

(c) [3] Use your loop invariant to determine the value of **answer** when the loop terminates for k=5.

3. [15] Show the values of the data structures *tree*, *min_wt*, and *closest* (as described in class) after each phase of the Dijkstra/Prim minimum spanning tree algorithm. The *phase* equals the number of vertices in the tree so far. The columns are numbered by the phase and the rows by the vertex numbers. Mark the edges in the MST.

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

start here



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

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

4. [10] Consider the following two schemes for storing *n* distinct keys in an array A.

   Scheme 1: Keys are stored in increasing order in A[0] to A[n-1].

   Scheme 2: A min-heap (minimum on top) is implemented in A[0] to A[n-1].

   How long does it take to perform the following operations in the worst case (Use "Θ" notation)?
   After each operation, the data structure must be maintained.
   Only algorithms valid under the comparison model should be considered.

Operation	Scheme 1	Scheme 2
1. Find the minimum		
2. Delete the minimum given its index in A		
3. Find the maximum		
4. Delete the maximum given its index in A		
5. Print out the keys in sorted order		

5. Give detailed pseudocode for weighted union (use the number of nodes in the component as the weight) and collapsing find.

(a) [5] W-UNION(u, v, parent)	(b) [5] C-FIND(u, parent)

(c) [5] Apply your algorithms from parts (a) and (b) to update the parent array given below to show the effects of calling W-UNION(5, 7, parent) where the W-UNION uses C-FIND.

i	0	1	2	3	4	5	6	7	8	9	10	11
parent[i]	9	0	9	5	-8	2	-5	10	0	4	6	6

After W-UNION(5, 7, parent):

i	0	1	2	3	4	5	6	7	8	9	10	11
parent[i]												

6. You are given an array of order n as input, $A[0..(n-1)]$. Your aim is to implement Quicksort using this array as the data structure. Suppose that instead of implementing Quicksort recursively, a stack is used to keep track of subproblems that need to be sorted. Entry [a,b] on the stack means that A[a] .. A[b] is a subproblem that needs to be sorted. The *size* of a subproblem is the number of elements that need to be sorted, and this equals b-a+1 for a subproblem [a, b]. Suppose that the two subproblems after a pivot are always placed on the stack so that the bigger subproblem is placed on the stack and then the smaller one.

(a) [5] When does the algorithm use a maximum amount of stack space?

(b) [10] Set up and solve a recurrence which gives the maximum amount of stack space that is required as per part (a). You may assume that $n = 2^k - 1$ for some integer k.

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

The End.