UNIVERSITY OF VICTORIA
EXAMINATIONS DECEMBER 2003
COMPUTER SCIENCE 225 (F01)

NAME: ______________________________ STUDENT NO. ______________

INSTRUCTOR: Frank Ruskey SECTION: F01

DURATION: 3 Hours

TO BE ANSWERED ON THE PAPER

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 10 PAGES.

NOTES: (0) CLOSED BOOK EXAM; NO NOTES OR CALCULATORS ALLOWED EXCEPT FOR A ONE PAGE "CHEAT SHEET," (1) ANSWER ALL QUESTIONS, (2) THERE ARE A TOTAL OF 96 MARKS, (3) SCRATCH PAPER IS AVAILABLE FROM THE INVIGILATORS, BUT NOTE THAT THERE IS BLANK PAGE AT THE END.

Page	Possible marks	Actual marks
2	20	
3	11	
4	8	
5	15	
6	9	
7	8	
8	12	
9	13	
Total	96	

1. Short answer questions. Total of 20 marks.

   (a) True or false? In the worst case the UNION-FIND algorithm with balancing and compression performs the FIND operation in $O(\log^* n)$ time?

   ANSWER: ______________________

   (b) How many binary heaps are there with the keys 1,2,3,4,5?

   ANSWER: ______________________

   (c) Give an example of two distinct insertion sequences that result in the same binary search tree. Make your example as small as possible.
   ANSWER: ____________ and ____________

   (d) What is the minimum height of an n-node 2-3-4 tree? Be as accurate as possible.

   ANSWER: ______________________

   (e) How many key comparisons are done in merging together two binomial queues, one with 40 elements and the other with 43 elements?

   ANSWER: ______________________

   (f) In hashing, what is the load factor of a table of size 100 that contains 60 keys?

   ANSWER: ______________________

   (g) In hashing, suppose that we start with a table of size one, do only insertions and lookups, and double the table size each time the load factor is 0.5. If 2^N items are in the final table, then *about* how many insertions did we perform?

   ANSWER: ______________________

   (h) Explain why we often use the expressions $O(E \log E)$ and $O(E \log V)$ interchangeably.

   ANSWER: ____

   (i) True or false? If a DFS search of a directed graph has no back edges, then the graph has no cycles?

   ANSWER: ____________

   (j) On a graph, what is the best method to use for finding a path from s to t that uses the least number of edges?

   ANSWER: ______________________

2. Draw the (directed) graph with the following adjacency lists. [2 marks]

   0: $\rightarrow 1$
   1: $\rightarrow 0 \rightarrow 2$
   2: $\rightarrow 3$
   3: $\rightarrow 1 \rightarrow 0$

3. Write the java code for a recursive method `LC` that returns the number of (non-null) left branches in a binary tree. Assume the nodes of the binary tree come from `class Node { Node left; Node right; }` . [3marks]

```
int LC ( Node p ) {









}
```

4. What is the best sorting method in the following situations? [4 marks]

   Scenario A: Sort a huge number of tiny files (independent of each other) Application: Daily customer transaction records

   Scenario B: Sort a file of huge records with tiny keys Example application: Reorganize your MP-3 files.

   Scenario C: Sort a huge randomly-ordered file of small records. Application: Process transaction record for a phone company

   Scenario D: Sort a file that is already almost in order. Applications: Re-sort a huge database after a few changes; Double-check that someone else sorted a file.

   Choices for sorting methods:

   0. bubble sort
   1. selection sort
   2. a system sort, such as quicksort
   3. insertion sort

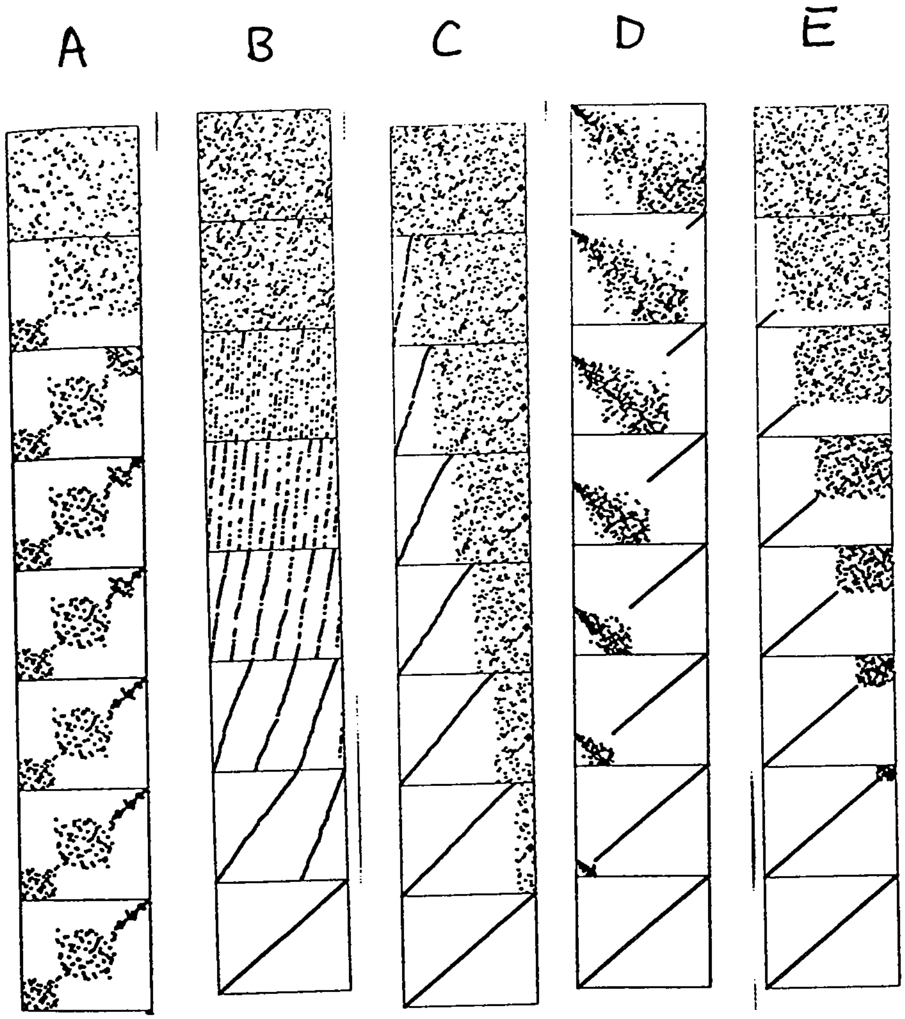
scenario	**best method (0,1,2,3)**
A	
B	
C	
D	

5. What is the result after the book's partitioning algorithm (as used in quicksort) is applied to the following array? Assume that the last element of the array is the partitioning element. [2 marks]

before	9	8	3	2	6	1	4	5
after								

6. Snapshots of five sorting methods are illustrated below. Identify each algorithm. [5 marks]

Snapshot	Name of sorting algorithm
A	
B	
C	
D	
E	



7. For each recurrence relation, name an algorithm analysis in which it occurred. [3 marks]

$C_N = N + 1 + \frac{1}{N}\sum_{1\le k\le N}(C_{k-1} + C_{N-k})$ ____________________

$C_N = N + 2C_{N/2}$ ____________________

$C_N = 1 + C_{N/2}$ ____________________

8. Perform a DFS of the graph whose adjacency lists are shown below. Denoted edges as ordered pairs (v, w). [10 marks]

   Which edges are cross edges? ______________________________
   Which edges are tree edges? ______________________________
   Which edges are down edges? ______________________________
   Which edges are back edges? ______________________________
   What is the preorder (DFS) numbering of the vertices (use the table)?
   What is the postorder numbering of the vertices (use the table)?
   List the vertices in each strongly connected component

   ____

   List the topological sort (ignore back edges) derived from the postorder numbering

   ____

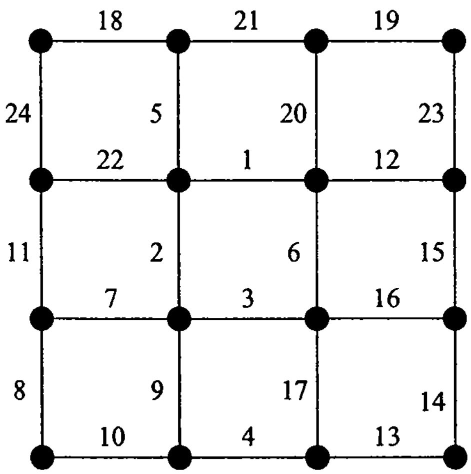
ord								
post								
vertex	0	1	2	3	4	5	6	7

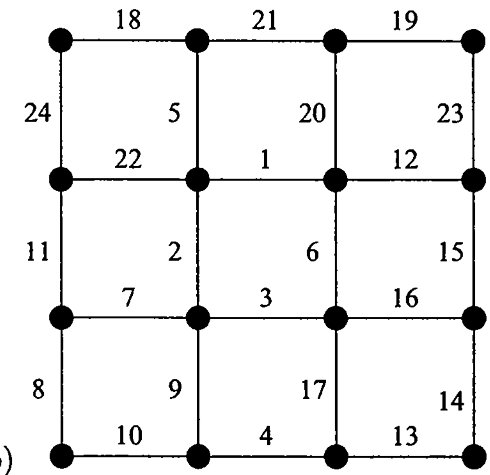
0: $\rightarrow$ 3
1: $\rightarrow$ 2 $\rightarrow$ 7
2: $\rightarrow$ 5 $\rightarrow$ 7
3:
4: $\rightarrow$ 6 $\rightarrow$ 0
5:
6: $\rightarrow$ 4
7: $\rightarrow$ 1 $\rightarrow$ 5

9. Depth-first search can be used to classify the edges of a graph as tree, cross, down, or back. This classification can be done as the DFS is proceeding if the preorder and postorder numbering of each vertex is also being computed. Suppose that the preorder numbering is stored in an array `ord` and the postorder numbering in an array `post`. Suppose that the current call is DFS(v) and that the edge (v, w) is being processed. Assume that the arrays `ord` and `post` are initialized to `-1`. In the table below show the condition (in terms of the `ord` and `post` values) that you would test in order to correctly classify each edge. [5 marks]

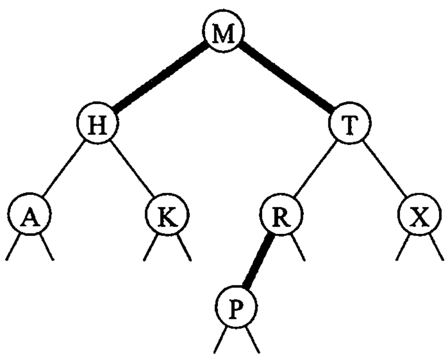
edge type	condition to test
tree	
back	
down	
cross	

10. Show the spanning tree after 11 edges have been added: (a) Using Prim's algorithm, (b) Using Kruskal's algorithm. [4 marks]

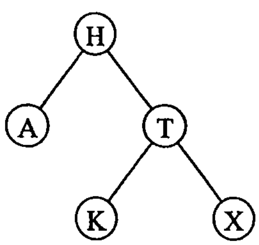
(a) 

(b) 

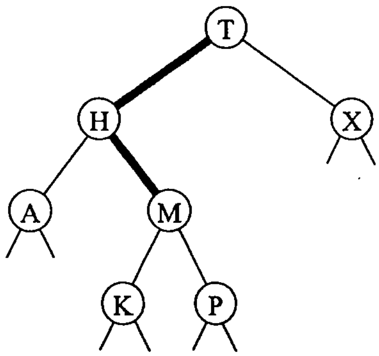
11. Draw the 2-3-4 tree that corresponds to the following red-black tree. [2 marks]



12. What tree results if the root insertion method is used to insert I into the following binary search tree? [3 marks]



13. (a) Perform two rotations on the following tree so that it will be a valid red-black tree. Show the tree after each rotation and indicate which edge is being rotated. (b) Under the right-child left-sibling rule show the ordered forest to which the (original) tree corresponds. [4 marks]



ordered forest:

after 1st rotation:

after 2nd rotation:

14. Show the successive steps of an LSD radix sort of the following data. [4 marks]

initial	pass 1	pass 2	pass 3
210			
222			
001			
020			
011			
111			
210			
012			

15. Below is the book's proof of the correctness of Kosaraju's algorithm for finding the strongly connected components of a graph. Each time there is a choice cross out the incorrect choice and circle the correct choice. Choices are indicated in italics and are separated by slashes; i.e., are of the form *choice A / choice B* . [6 marks]

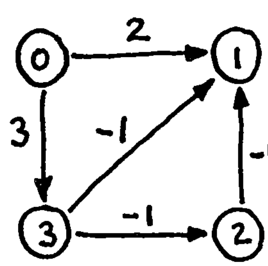
    If s and t are mutually reachable, they *will / will not* be in the same DFS tree because when the first of the two is visited, the second is *also visited / unvisited* and is reachable from the first and so will be visited *before / after* the recursive call to the root terminates.

    To prove the converse, we assume that s and t are in the same tree, and let r be the root of the tree. The fact that s is reachable from r (through a directed path of tree edges) implies that there is a directed path from s to r in the reverse digraph. Now, the key to the proof is that there must also be a path from r to s in the reverse digraph because r has a *lower / higher* postorder number than s (since r was chosen *first in / after* in the second DFS at the time when both were unvisited) and there is a path from s to r: If there were not path from r to s, then the path from r to s in the reverse would leave s with a *lower / higher* postorder number. Therefore, there are directed paths from s to r and from r to s in the digraph and its reverse: s and r are strongly connected. The same argument proves that t and r are strongly connected, and therefore s and t are strongly connected.

16. In justifying the various minimum spanning tree algorithms we used the red and blue rules and a colour invariant. [3 marks]

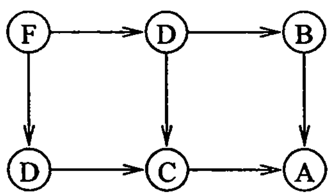
    State the colour invariant.

17. On the graph below Dijkstra's algorithm will fail. Explain why. Show the successive edge relaxations that the Bellman-Ford-(Moore) algorithm will make on that graph. The start node is 0. [3 marks]



18. Write the java code snippet necessary to set an array `cnt[n]` so that `cnt[v]` is the number of edges of the form (w, v); i.e., it is the number of edges that coming into v. Assume that the adjacency lists are referenced through an array `Node[] adj` with `class Node { int v; Node link; }`. The running time of your algorithm should be $O(V + E)$. [3 marks]

19. What is the lexicographically smallest topological sorting of the following graph? What is the lexicographically largest? Describe an $O(E + V \log V)$ algorithm for finding the lexicographically smallest topological sorting. Explain why your algorithm has the stated running time. NOTE: lexicographic order is like dictionary order. HINT: the previous problem may be helpful! [10 marks]



Lex smallest: ____________________

Lex largest: ____________________

This page intentionally left blank.

THE END