CSC 225 Final Exam: Summer 1997
Instructor: W. Myrvold

1. [12] 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 $O(n \log n)$ in the worst case is always faster than an algorithm which is $O(n^2)$ in the worst case.
**True** **False**

(b) It is possible to sort an array of $n$ numbers in $O(n \log n)$ time in the worst case using only $O(1)$ extra space.
**True** **False**

(c) Let $f$, $g$, and $h$ be functions from the natural numbers to the positive real numbers. Then if $g \in \Omega(f)$ and $g \in O(h)$, and $f \in O(h)$ then $g \in \Theta(h)$.
**True** **False**

(d) In order to find a minimum weight spanning tree, we might have to completely order the edges of the graph by edge weight. Thus under the comparison model, the time complexity of any algorithm for minimum spanning tree is $\Omega(m \log m)$ in the worst case, where $m$ is the number of edges.
**True** **False**

2. For this question, consider the recurrence defined by:

$$T(n) = 5 + T(n/2), T(8) = 6.$$

Assume that $n = 2^k$ for some integer $k$. Show all your work for full marks.

(a) [5] Solve this recurrence by repeated substitution to get a **sum**.

(b) [2] Give a closed formula for your sum from (a).

(c) [3] Prove by induction that your closed formula from (b) is a correct solution to the recurrence.

3. [12] Your job is to analyse various data structures for implementing a set. The sets under consideration can contain up to $n$ elements represented by the integers from zero up to $n-1$. Consider the following data structures for a set $S$:

(a) The elements of set $S$ are stored in a sorted array.

(b) The elements of set $S$ are stored in an unsorted linked list (no duplicates allowed).

(c) The elements of $S$ are indicated by a *characteristic vector* which is an array of size $n$ which has position $i$ set to 1 if element $i$ is in the set and 0 if it is not in the set.

Let $S$ be a set which has size $s$ and let $T$ be a set of size $t$. Give the **worst case** time complexities of the best algorithms for the following set operations as functions of $n$, $s$, and $t$.

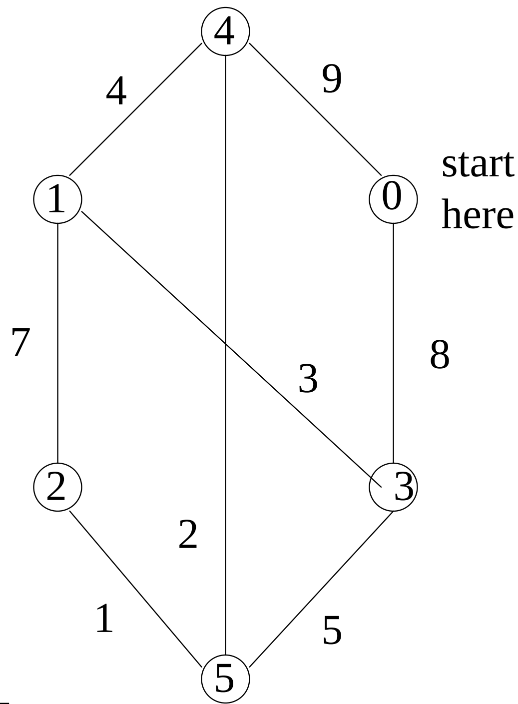
| Operation | (a) Sorted array | (b) Unsorted list | (c) Characteristic vector |
|---|---|---|---|
| Is $x$ in $S$? | | | |
| Add $x$ to $S$ (do not count time to check if $x$ in $S$) | | | |
| Print elements in $S$ in sorted order | | | |
| Create union of $S$ and $T$ | | | |

Questions 4-5 refer to the set operations as described for question #3. After you complete these questions, you may want to reconsider the answers you gave for question 3.

4. Suppose the $s$ elements of set $S$ are stored in an unsorted array and you are restricted to an algorithm which is valid under the comparison model.
   (a) [10] Give the pseudo code for an algorithm for traversing the set elements in sorted order which has optimal worst case time complexity.
   (b) [5] Analyze the worst case time complexity of your algorithm from (a) using Big Oh notation.
   (c) [10] Argue that your algorithm from part (a) is optimal in the worst case for the comparison model. You may assume that $\log_2(n!) \in \Theta(n \log n)$.

5. Suppose the $s$ elements of set $S$ are stored in an unsorted array and you are no longer restricted to the comparison model.
   (a) [5] Give pseudo code for an optimal algorithm for traversing the elements of $S$ in sorted order if $s$ is in $\Theta(n)$.
   (b) [3] Analyze the worst case time complexity of your algorithm from part (a) using Big Oh notation.
   (c) [2] Argue that your algorithm from part (a) is optimal.
   (d) [5] For which values of $s$ (expressed as a function of $n$) is the approach you describe for Question #4 asymptotically faster than the one you describe for Question #5?

6. [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 | | | | | | | |

| 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 | | | | | | | |

7. [12] Draw pictures to show the state of the weighted union (W-UNION), collapsing find (C-FIND) data structure after each of the four breakpoints in the following program. Use the number of nodes in each component as the weight.

**Break ties for W-UNION by choosing the element with the smallest label to be the new root.**
Note: W-UNION calls C-FIND.

W-UNION(1,2)
W-UNION(3,4)
W-UNION(5,1)
W-UNION(6,4)
W-UNION(2,3)

**BREAKPOINT (a)**

W-UNION(7,8)
W-UNION(9,10)
W-UNION(11,7)
W-UNION(12,10)
W-UNION(8,9)
W-UNION(10,4)

**BREAKPOINT (b)**

C-FIND(12)

**BREAKPOINT (c)**

W-UNION(6,13)

**BREAKPOINT (d)**