CS 2005 Techniques of Programming
Craig E. Wills
Given: Friday, Dec 16, 1994

WPI, B Term 1994
Exam 3 (100 pts)

NAME:______________________________

LAB TIME:______________________________

WPI BOX NUMBER:______________________________

This is a closed book (and notes) examination. Answer all questions on the exam itself. Take the number of points assigned to each problem and the amount of space provided for your answer as a measure of the length and difficulty of the expected solution. The exam totals 100 points.

SCORE:____________________________________

1. (12 points) Indicate whether the following statements are true or false (write True or False, not T or F).

__________ The merge sort algorithm does more computation to partition the list into two sublists while the merge of the two sorted sublists is trivial.

__________ When recursion can be used it is often better than iteration for coding an algorithm because it results in having to write less code.

__________ The output of a postorder traversal of an expression tree could be used directly as the input for a stack-based expression evaluator.

__________ A breadth-first traversal of a graph can easily be done using recursion.

2. (12 points) This question relates to project 3 on the mini-hypertext system.

   (a) The structure for storing a piece of text from an article allows three types of text pieces to be stored. What are these three types?

   (b) When the user issues the command to follow a reference given in `<>`'s what type of searching (linear, binary, hashing) is done? Answer all that apply.

   (c) When the user issues the command to goto an article based on its title what type of searching (linear, binary, hashing) is done? Answer all that apply.

   (d) When the user issues the command to add a title to the hot list what type of searching (linear, binary, hashing) is done? Answer all that apply.

3. (8 points) For a bubble sort of n numbers how many comparisons are made in worst case (use big oh notation)? How many swaps are performed (use big oh notation)? If the following list of numbers is sorted using bubble sort, show the contents of the list after each iteration of the algorithm.

```
4    8    7    6    1
```

4. (9 points) Show the output of applying the following set operations where *PrintSet()* prints the members of the set in order.

```
typedef unsigned int Set;
Set s, t;

InitSet(s);
InitSet(t);
AddMember(s, 3);
AddMember(s, 5);
AddMember(s, 9);
AddMember(t, 5);
AddMember(t, 4);
AddMember(t, 1);
AddMember(t, 9);
PrintSet(s);
PrintSet(t);
PrintSet(Union(s, t));
PrintSet(Intersect(s, t));
PrintSet(Difference(s, t));
DeleteMember(t, 5);
DeleteMember(t, 7);
PrintSet(t);
```

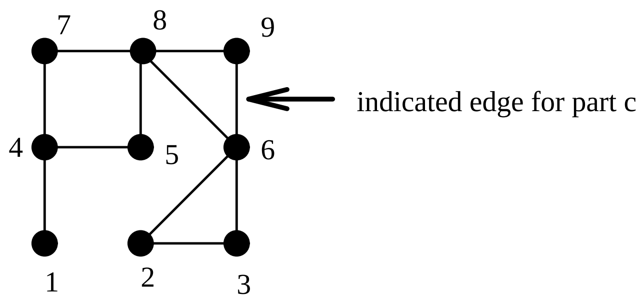
Now assume the `Set` abstraction is implemented as a bit vector where the setting the ith bit indicates the value i is in the set. Show how the *AddMember()* operation is defined.

5. (12 points) Show (pictorially) the binary search tree that is built from the following set of numeric ids when added to the tree in the order shown. The first number should be the root of your tree.

```
5011
2021
3033
6544
3237
7777
```

What would the tree look like if the node `2021` were deleted from the tree?

6. (12 points) Consider the following graph.



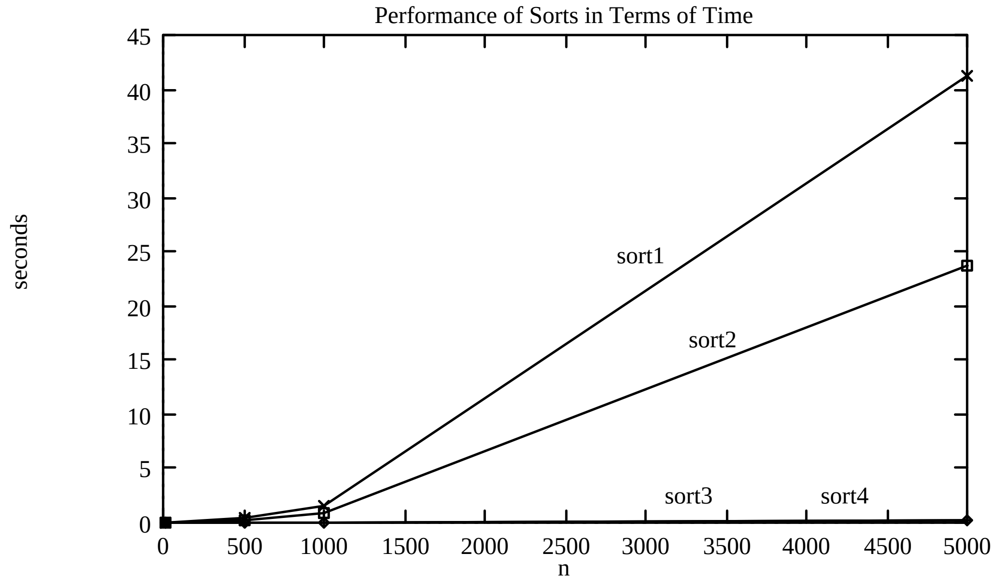
(a) Is the graph undirected or directed?

(b) Does the graph contain a cycle?

(c) How would the indicated edge be labeled?

(d) Print the order in which nodes are visited for a depth-first search starting from node 1. If choosing among a set of vertices for traversal always choose the lowest numbered vertex for traversal.

7. (12 points) In lab 7 you timed and graphed the execution time for four sorting algorithms (bubble, selection, merge and quick). The following graph shows typical results for the execution times of the four algorithms on various values of n. Each line is labeled (sort1 to sort4) with the lines for sort3 and sort4 both very close to the x-axis and indistinguishable from each other.



Indicate which sort corresponds to each performance line in the graph (circle a sort) for parts a-d and answer the questions in parts e and f.

(a) The line labeled sort1 is the performance for bubble, selection, merge, or quick sort?

(b) The line labeled sort2 is the performance for bubble, selection, merge, or quick sort?

(c) The line labeled sort3 is the performance for bubble, selection, merge, or quick sort?

(d) The line labeled sort4 is the performance for bubble, selection, merge, or quick sort?

(e) If sort1 has the worst performance, why consider using it at all?

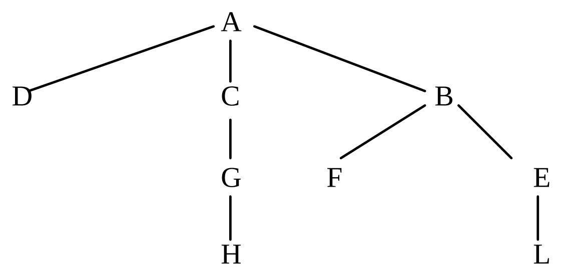
(f) If sort3 and sort4 give essentially the same performance why choose one over the other?

8. (8 points) What value is returned by the function call `Mystery("foobar")` where `Mystery` is defined below? In general, for any string passed to it what will the function return?

```
int Mystery(char *pch)
{
    if (*pch == '\0')
        return(0);
    else if (strlen(pch)%2 == 1)
        return(1 + Mystery(pch + 1));
    else
        return(Mystery(pch + 1));
}
```

9. (15 points) Using the tree structure declared in project 4, and duplicated below, write a recursive function *CountChildren()*, which is passed the root of a tree and a node value and returns the count of children for that node in the tree. Thus for the sample tree shown your function should return a value of 3 if passed `'A'` and return a value of 2 if passed `'B'`. Return a -1 if the node value is not found. The function must be recursive and *not* use any other functions written for project 4.

```
typedef struct treetag {
    char chNode;                /* value of the node */
    struct treetag *pTreeLeft; /* pointer to leftmost child */
    struct treetag *pTreeNext; /* pointer to next sibling */
} TreeNodeType;
```




```
int CountChildren(TreeNodeType *pTreeRoot, char chValue)
    /* your code goes here */
```