CMIS 240.4031: Data Structures and Abstraction

University of Maryland, University College – Spring 1998

Final Exam

Directions:

- Put your name on each page of the exam.
- Write directly on the exam. (Extra paper is available up front.)
- Show your work.
- This exam is open-book, open-notes. You may use any non-living, non-electronic resource that you have available.
- The exam starts at 7:00 and ends ***PROMPTLY*** at 9:55. There are students who cannot stay late, and it would not be fair to allow anyone to continue after that time.
- Note: For part I of the exam you are not required to answer all problems. If you answer more than the indicated number of problems, only the first N problems will be graded regardless of whether they are right or wrong.

- Part I: Short answer

This section contains twenty-five (25) problems. Select and answer fifteen (15). Each problem that you answer is worth five (5) points.

1. Write the first few lines of this function so that it uses the assert facility to check its precondition: (you do not have to write the whole function!)

```
void exam(int i)

// Precondition: i is not equal to 42.
...
```

2. Why is the order of an algorithm generally more important than the speed of the processor?

3. Convert each time formula to the best possible big-O notation. Do not include any spurious constants in your big-O answer.

Time Formula	Big-O Equivalent
10n	
$2n^2$	
3 times log (base 2) of n	

$2n^2 + 10n$	

4. What is an automatic default constructor, and what does it do?

5. What is an inline member function, and when might one be used? Give a small example as part of your answer.

˝

6. Suppose that you define a new class called Foo. For two Foo objects x and y, you would like the expression x+y to be a new Foo object. What is the prototype of the function that you must write to enable expressions such as x+y?

7. What happens if you call new but the heap is out of memory?

8. What are the steps to inserting a new item at the head of a linked list? Use one short English sentence for each step.

9. Suppose that p is a pointer to a Node in a linked list, and *p is not the tail node. What are the steps to removing the Node after *p? Use one short English sentence for each step.

10. Implement the following function as a new function for the linked list toolkit. (Use the usual Node definition with member variables called data and link.)

```
bool data_is_on(Node* head_ptr, Node* p);
// Precondition: head_ptr is the head pointer of a linked list
// (which might be empty, or might be non-empty). The pointer p
// is a non-NULL pointer to some Node on some linked list.
// Postcondition: The return value is true if the data in *p
// appears somewhere in a data field of a node in head_ptr's
// linked list. Otherwise the return value is false.
// None of the nodes on any lists are changed.
```

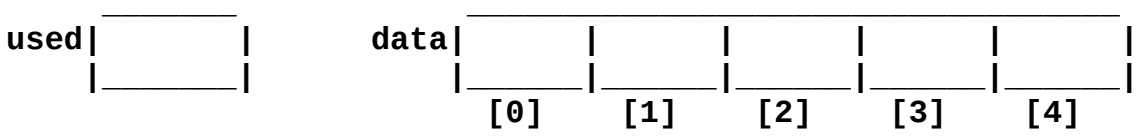
11. Tell me about one of the List operations that is more efficient because the class keeps a tail pointer as a private member variable. Provide a specific example showing why the operation would be less efficient without the tail pointer.

12. Write one or two short sentences to clearly explain the advantage of template functions.

13. I am going to execute this code with THREE pushes and ONE pop:

```
Stack<int> s;
s.push(1);
s.push(2);
s.push(3);
cout << s.pop( );
```

Suppose that s is represented by a (initially empty) partially filled array. Draw the state of the private member variables of s after the above code:



14. Complete the body of this function. Use a Queue of characters to store the input line as it is being read.

```
size_t counter( )
// Precondition: There is a line of input waiting to be read
// from cin.
// Postcondition: A line of input has been read from cin, up to
// but not including the newline character. The return value of
// the function is the number of times that the LAST character
// of the line appeared somewhere in this line.
// EXAMPLE Input: ABBXDXXZX
// The value returned by counter would be 4 for this input since
// there are 4 X's in the input line.
{
  size_t answer = 0;
  Queue<char> q;
```

15. What is the importance of the stopping case in recursive functions?

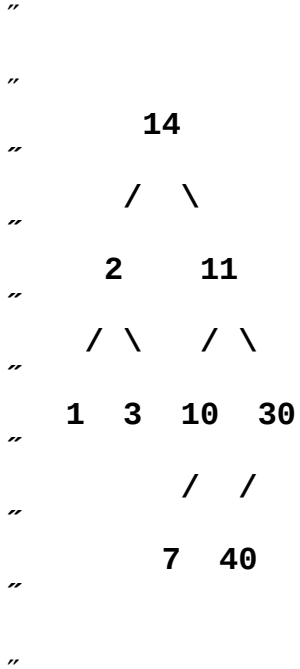
16. Write a recursive function that has one parameter which is a size_t value called x. The function prints x asterisks, followed by x exclamation points. Do NOT use any loops. Do NOT use any variables other than x.

17. Write a function with one positive int parameter called n. The function will write 2^n-1 integers (where ^ is the exponentiation operation). Here are the patterns of output for various values of n:

```
n=1: Output is: 1
n=2: Output is: 1 2 1
n=3: Output is: 1 2 1 3 1 2 1
n=4: Output is: 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1
```

And so on. Note that the output for n always consists of the output for n-1, followed by n itself, followed by a second copy of the output for n-1.

18. Here is a small binary tree:



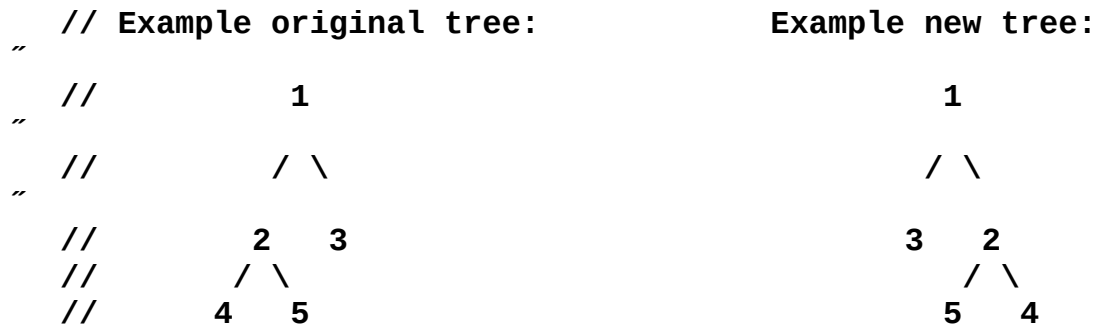
Circle all the leaves. Put a square box around the root. Draw a star around each ancestor of the node that contains 10. Put a big X through every descendant of the node the contains 10.

19. Consider this BinaryTreeNode definition:

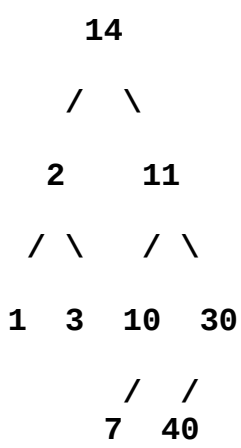
```
struct BinaryTreeNode
{
    int data;
    BinaryTreeNode *left;
    BinaryTreeNode *right;
};
```

Write a function to meet the following specification. Check as much of the precondition as possible. No recursion is needed.

```
void flip(BinaryTreeNode* root_ptr)
// Precondition: root_ptr is the root pointer of a non-empty
// binary tree.
// Postcondition: The tree is the mirror image of its original
// tree.
```


// Example original tree:
Example new tree:
//
1
1
//
/ \
/ \
//
2 3
3 2
//
/ \
/ \
//
4 5
5 4

20. Here is a small binary tree:



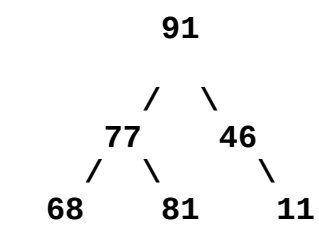
Write the order of the nodes visited in:

A. An in-order traversal:

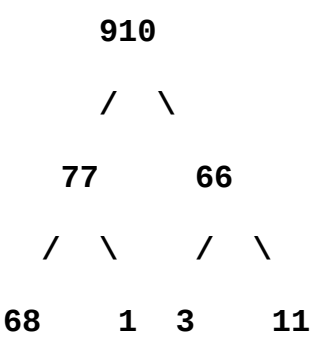
B. A pre-order traversal:

C. A post-order traversal:

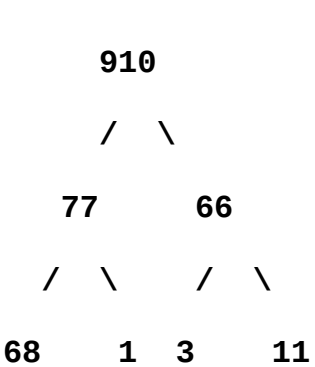
21. Give two different reasons to explain why the following binary tree is not a heap:



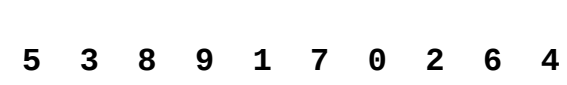
22. Draw a new heap that is created by inserting 82 into the following heap:



23. Draw a new heap that is created by deleting item 77 from the following heap:



24. Here is an array of ten integers:



Draw this array after the FIRST iteration of the large loop in a selection sort (sorting from smallest to largest).

Draw this array after the SECOND iteration of the large loop in a selection sort (sorting from smallest to largest).

Draw this array after the THIRD iteration of the large loop in a selection sort (sorting from smallest to largest).

25. Implement the following function:

```
   void merge(int data[ ], size_t n1, size_t n2);

   // Precondition: The first n1 elements of data are sorted, and
   // the next n2 elements of data are sorted (from smallest to
   // largest).

   // Postcondition: The n1+n2 elements of data are now completely
   // sorted.
```

Part II: Multiple choice

This section contains fifteen (15) problems. Answer all of them. Each problem that you answer is worth one (1) point.

1. Why is writing easily modifiable code important?
   A. Easily modifiable code generally has a quicker run time.
   B. Most real world programs require change at some time.
   C. Most text editors make it easy to modify code.
   D. Several people may be writing the same function at the same time.

2. What ANSI C function is used to stop the program execution when a precondition is not met.
   A. assert();
   B. exit();
   C. return();
   D. void();
   ″

″

3. Why is it important to test boundary values when testing programs?
   A. Calculating by hand, it's easy to find the right answers for boundary values.
   B. Debuggers are easier to use when testing boundary values.
   C. In practice, a large proportion of errors arise from boundary values.
   D. The correct execution of a function on all boundary values proves a function is correct.

4. Suppose that the Foo class does not have an overloaded assignment operator. What happens when an assignment a=b; is given for two Foo objects?
   A. The automatic assignment operator is used
   B. The copy constructor is used
   C. Compiler error
   D. Run-time error
   ″

″

5. When should you use a const reference parameter?
   A. Whenever the data type might be many bytes.
   B. Whenever the data type might be many bytes, the function changes the parameter within its body, and you do NOT want these changes to alter the actual argument.
   C. Whenever the data type might be many bytes, the function changes the parameter within its body, and you DO want these changes to alter the actual argument.
   D. Whenever the data type might be many bytes, and the function does not change the parameter within its body.

6. Which kind of functions can access private member variables of a class?
   A. Friend functions of the class
   B. Private member functions of the class
   C. Public member functions of the class
   D. All of the above can access private member variables
   E. None of the above

7. Suppose that Foo is a new class and you want to declare an array of 10 Foo objects. Which constructor will be used to initialize the 10 Foo components of the array?
   A. Only the copy constructor can be used
   B. Only the default constructor can be used
   C. Any constructor can be used

8. Suppose that you are working on a machine where arrays may have up to 4,000,000,000 items. What is the guaranteed range of the size_t data type?
   A. Negative 4,000,000,000 to positive 4,000,000,000.
   B. Zero to positive 32,767
   C. Zero to positive 65,535
   D. Zero to 4,000,000,000
   E. Zero to 8,000,000,000

9. Consider the following statements:

```
int *p;
int i;
int k;
i = 42;
k = i;
p = &i;
```

After these statements, which of the following statements will change the value of i to 75?

   A. k = 75;

B. *k = 75;
C. p = 75;
D. *p = 75;
E. Two or more of the answers will change i to 75.

˝

10. In which location do dynamic variables reside?
    A. The code segment.
    B. The data segment.
    C. The heap.
    D. The run-time stack.

11. When a class uses dynamic memory, what member functions should be provided by the class?
    A. The assignment operator.
    B. The copy constructor.
    C. A destructor.
    D. All of the above.

12. What is the fundamental difference between a struct and a class?
    A. By default the members of structs are private.
    B. By default the members of structs are public.
    C. Structs can only contain data members.
    D. Structs must contain a pointer member variable.

13. Suppose that p is a pointer variable that contains the NULL pointer. What happens if your program tries to read or write *p?
    A. A syntax error always occurs at compilation time.
    B. A run-time error always occurs when *p is evaluated.
    C. A run-time error always occurs when the program finishes.
    D. The results are unpredictable.

14. What is the primary purpose of template functions?
    A. To allow a single function to be used with varying types of arguments
    B. To hide the name of the function from the linker (preventing duplicate symbols)
    C. To implement container classes
    D. To permit the use of the debugger without the -gstabs flag

15. Suppose Bag is a template class, what is the syntax for declaring a Bag b of integers?
    A. Bag b;
    B. Bag<int> b;
    C. Bag of int b;
    D. int Bag b;

Part III: Roll your own

As discussed last week, pose a question that was not covered on this exam, and answer it. Your question must be related to C++. This part is worth 10 points.