Midterm 1

CS 14 - Data Structures

April 19, 2013

By taking this exam, I affirm that all work is entirely my own. I have not cheated in any way.

Signature: ______________________________

Printed Name: ______________________________

1. (3 pt) What does $f = \Omega(g)$ mean? Give an exact definition.

2. (1pt) Give a *tight* upper bound on the function:

$$f(x) = \frac{45x^3 + x^2 \log x + 10}{x}$$

3. (3pt) Simplify the following:

   (a) $\Omega(n) * \Omega(\log n)$

   (b) $\Theta(n \log(n^3)) + \Theta(n^2)$

   (c) $O(n(1 + \log_5 n) + \log_5 2^n)$

4. (3pt) Circle the bigger function; if they are the same, circle "equal."

   (a) $\Theta(1)$ or $\Theta(2)$ or equal

   (b) $\Omega(n \log n)$ or $\Omega(n \log n^2)$ or equal

   (c) $O(1.1^n)$ or $O(n^{4000})$ or equal

5. (3pt) Give three examples of stacks

   (a)

   (b)

   (c)

6. (3pt) Give three examples of queues

   (a)

   (b)

   (c)

7. (3pt) List three ways of selecting the pivot in quicksort

   (a)

   (b)

   (c)

8. (2pt) Solve the RPN equation: $5\ 4 + 3\ 4 + *$

9. (4pt) Calculate the following bitwise operations. The numbers are displayed in binary.

   (a) 11111111 & 00010111

   (b) 00100010 | 10100000

10. For the **quicksort** algorithm:

    (a) (5pt) Write the algorithm using recursion; you may assume the existance of a function to concatenate two containers.

    (b) (3pt) Write the best, worst, and average case runtimes in Θ notation.

    (c) (2pt) Name one advantage and one disadvantage of quicksort in comparison to merge sort.

11. For the **binary search** algorithm:

    (a) (5pt) Write the algorithm using recursion

    (b) (3pt) Write the best, worst, and average case runtimes in Θ notation.

    (c) (2pt) Does it make since to call binary search on a singly linked list data structure? Why or why not?

12. The class below implements a queue using two stacks.

(a) (10pt) Implement the enqueue and dequeue functions. You may assume that stacks have member functions for push, pop, and size.

```
template <typename T> class Queue
{
public:
    Queue(void);
    ~Queue(void);

    void enqueue(const T& node);
    T dequeue();

private:
    Stack<T> stack1;
    Stack<T> stack2;
};

template<typename T> void Queue<T>::enqueue(const T& element) {










}

template<typename T> T Queue<T>::dequeue() {










}
```

(b) (5pt) What are the best and worse case run times of *both* functions in terms of the number of elements n in the queue? Explain in words what inputs cause the best and worst case.

13. (20pt) Solve the recurrence relations. Report your answer in Θ notation.

(a) $T(n) = 10T\left(\frac{n}{10}\right) + 2n$; $T(1) = 1$

(b) $T(n) = T\left(n-1\right) + T(n-2) + 1$; $T(1) = 1$

14. Given the function below:

```
void oddkiller(vector<int> &v)
{
    for (vector<int >::iterator i=v.begin(); i!=v.end(); i++) {
        if (*i%2==1) {
            i=v.erase(j);
        }
    }
}
```

(a) (5pt) Provide best and worst case runtime analysis in terms of the size n of the input vector.

(b) (10pt) Rewrite the function so that instead of using a for loop, it uses recursion. You do not have to use iterators, but if you use them correctly you will get 2pt of extra credit.

Review questions

15. (3pt) What are the “big 3” functions in c++?

16. (1pt) When must we write the “big 3” functions?

17. (1pt) What makes a class an “abstract base class”?

Extra credit

18. (1pt) What does STL stand for?

19. (1pt) What does ADT stand for?

20. (1pt) In C++, what is the :: operator called?