FSU COP 4530 / CGS 5425 (Spring 2009)
Data Structures, Algorithms, and Generic Programming

Due Feb 13

Assignment 3: Max points: 50

Please show steps or give justifications for all your answers, unless we specify otherwise.
Each question is worth 5 points.

1. Assume that the time complexity of an algorithm on input of size n is $3n^3$. If the algorithm takes s seconds to execute on some computer, on an input of size n, then how many seconds will it take on an input of size $4n$?

2. Let the time complexity of an algorithm be: $1000n^3 \log^2 n + 2n^4 \log n + 3n^4 - 8n^2 + - 1$. Give the asymptotic time in big-Oh notation. (You need not show steps.)

3. Show that $2n^3 + 4n$ is $O(n^3)$ directly from the definition of big-O, by finding appropriate constants c and N.

4. If $f_1(n) = O(g_1(n))$ and $f_2(n) = O(g_2(n))$ then prove, using the definition of Big-O, that $f_1(n) \times f_2(n) = O(g_1(n) \times g_2(n))$.

5. If $f_1(n) = O(g_1(n))$ and $f_2(n) = O(g_2(n))$, then show, using a counterexample, that $f_1(n)/f_2(n)$ is not necessarily $O(g_1(n)/g_2(n))$.

6. Draw the state of a self-organizing doubly linked list (count method) after the following operations. Show the links and counts too.

```
   insert(a); insert(b), insert(c), search(b), search(b), search(c)
```

7. Which container, among those we have discussed (*vector*, *sorted vector*, *singly linked list*, or *doubly linked list*), will you use to store a set of student records? Assume that the following operations are performed: `insert` a record (making sure that multiple records are not inserted for the same student), `delete` the record of a student, or `search` for the record of a student. Justify your answer by comparing the time and space complexities of each container. Specify any assumptions that you make, such as assumptions regarding the relative likelihood of different operations.

8. Give the average number of comparisons in sequential search, with the following probability distribution. $\text{Prob}(\text{searching for element at location } 0) = 1/3$, $\text{Prob}(\text{searching for element } i) = 1/(3[n-1])$, $1 \leq i < n$. $\text{Prob}(\text{element not found}) = 1/3$.

9. In vector `push_back`, instead of doubling the capacity each time that the current capacity is exceeded, assume that we make it *four times* the current capacity. What is the amortized time complexity for n `push_back`s, if the initial capacity was 1? Show all steps.

10. How many times is the statement with `cout <<` … executed, in the following pseudocode? Derive the exact number, showing all steps.

```
   M = 1
   for i = 0 to n-1 {
       for j = 1 to M
           cout << i << j << endl;
       M *= 2
    }
```