Homework 3

Out: 10.3.16
Due: 10.12.16

1. [Recurrences, 28 points]
   Give the tightest asymptotic bounds you can for the following recurrences and provide a short explanation for your solution.
   a. $T(n) = 16T(n/4) + 2n^2$
   b. $T(n) = 7T(n/2) + n^2$
   c. $T(n) = T(n-1) + n$
   d. $T(n) = 2T(n/2) + n^4$
   e. $T(n) = 2T(n/4) + 3\sqrt{n}$
   f. $T(n) = 2T(n/2) + n/\lg n$
   g. $T(n) = n(T(n/2)^2)$

2. [Recurrences II, 32 points]
   For each of the following functions, provide:
   - A recurrence T(n) that describes the worst-case runtime of the function in terms of n as provided (i.e. without any optimizations)
   - The tightest asymptotic upper and lower bounds you can find for T(n)

   a.
   ```
   int A(int n) {
     int sum=0;
     for (i=0; i<n; i++)
       for (j=0; j<n*n; j++)
         sum++;
   }
   ```

   b.
   ```
   int B(int n) {
       if (n == 0) return 1;
       else return 2*B(n-1);
   }
   ```

   c.
   ```
   int C(int n)
   {
       if (n == 0) return 1;
       if (C(n/2) >= 5)
               return C(n/2) + 5;
       else
               return 5;
   }
   ```

   d. *(see next page)*

```
int D(int n) {
  int sum = 1;
  if (n > 1000)
        for (int i=1; i<n; i++)
            sum += Math.exp(i);
        return sum;
  else
        int tmp = n;
        for (int i=1; i<n ; i++)
            for (int j=0; j<n; j++)
                if (tmp < (i*j))
                  sum++;
 return D(n/2) * (sum %2);
}
```

3. [Templates, 40 points]
A singly-linked list is implemented with a head node, but no tail node, and contains a pointer to the head node. Design a class to implement this linked list, that includes the following methods:
    - LinkedList – Constructor of an empty linked list.
    - ~LinkedList – Destructor of a linked list.
    - size – Returns the size of the linked list (i.e. the number of nodes).
    - contains – Tests if a value x is contained in the linked list, returns bool.
    - add – Inserts an integer value x if it is not already contained in the linked list.
    - remove – Removes an integer value x if it is contained in the linked list.
    - print – Prints the content of the linked list.

   For each of the following tasks, your code must compile and run on the lab computers. Make sure to write your name in a comment at the top of the main file, along with any special instructions on how to compile your code.

   Use the provided *makefile* to compile your code: Place it in the same directory with your code and type 'make a' for part (a), or 'make b' for part b. You may modify the *makefile* as needed and submit it with your code.

   a. The files *LinkedList.h* and *main_a.cpp* are provided. Implement and submit a *LinkedList.cpp* file that will compile with the provided files (do not modify the provided files), along with a *makefile*.

   b. Modify your code, such that a node is templetized, and can store any pre-defined object and not only an int. You will need to modify and submit LinkedList.h and possibly also LinkedList.cpp. Your code should compile with the provided *main_b.cpp* file. Also submit a *makefile*.
   Note: While the declaration and implementation of classes are typically split between .cpp and .h files, some compilers cannot handle templates in separate files, so you may want to include both the declaration and implementation in a single .h file.