Exam 1
Computer Science 751
Lehman College– CUNY
Thursday, 17 October 2002

NAME (Printed) ____________________________
NAME (Signed) ____________________________
Login ____________________________

Please show all your work and circle your answers. Your grade will be based on the work shown.

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td>TOTAL</td>
<td>100</td>
</tr>
</tbody>
</table>
Useful Formulas

\[ \sum_{i=1}^{n} i = \frac{n(n+1)}{2} \]
\[ \sum_{i=1}^{n} \frac{1}{i} = \ln n + O(1) \]
\[ \sum_{i=1}^{n} x^i = \frac{x^{n+1}-1}{x-1} \]
\[ \sum_{i=1}^{\infty} \frac{1}{i} = \ln n + O(1) \]
\[ \sum_{i=0}^{\infty} ix^i = \frac{x}{(1-x)^2} \]
\[ e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots \]
\[ \lim_{n \to \infty} (1 + \frac{x}{n})^n = e^x \]
\[ n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n (1 + O(\frac{1}{n})) \]
\[ n! = o(n^n) \]
\[ n! = \omega(2^n) \]
\[ \log(n!) = \Theta(n \log n) \]
\[ \binom{n}{k} = \frac{n!}{k!(n-k)!} \]
\[ \binom{n}{k} \leq \binom{en}{k} \]
\[ E[X] = \sum_x x Pr[X = x] \]
1. True or False (2 point each):
   
   (a) \( \lg n = o(n^2) \).
   (b) \( \lg n = O(n^2) \).
   (c) \( 3n^2 + 2n = \omega(n) \).
   (d) \( 3n^2 + 2n = \Omega(n) \).
   (e) \( n! = \Omega(2^n) \).
   (f) \( \lg(n!) = \Theta(\lg(n^n)) \).
   (g) \( f(n) = o(g(n)) \) implies \( f(n) = O(g(n)) \).
   (h) \( f(n) = \Omega(g(n)) \) implies \( f(n) = \Theta(g(n)) \).
   (i) \( f(n) = \Theta(g(n)) \) implies \( f(n) = O(g(n)) \).
   (j) \( f(n) = \Theta(g(n)) \) implies \( f(n) = \omega(g(n)) \).

2. Assume that every statement takes a constant \( c \) time. Give tight bounds on the order of growth and justify your answer:
   
   (a) What is the output, assuming the following piece of code is embedded in a complete and correct program:
       
       ```cpp
       for ( int i = 5; i > 0; i--)
       {
           for ( int j = 0 ; j < i; j++)
               cout << '*';
           cout << endl;
       }
       ```

   (b) Assume \( A \) is an array of length \( n \):
       
       ```cpp
       FIND-MAX(A)
       1 max <- - infinity
       2 for i <- 1 to n
       3      do if A[i] > max
       4          then max <- A[i]
       5 return max
       ```

   (c) Assume \( A \) is an array and the function COMBINE takes \( \Theta(n) \) on a sublists \( A[p..r] \) and \( A[r+1..p] \) of combined length \( n \):
       
       ```cpp
       MSORT(A,p,q)
       1 if ( q - p > 1)
       2      do MSORT(A,p, q/2);
       3      MSORT(A,q/2+1,q);
       4      COMBINE(A,p,q/2,r);
       ```
3. Assume $A$ is an array storing a heap and $k$ is a key:

\[
\text{HEAP-INSERT}(A,k)
\]
\[
\begin{align*}
1 & \text{ heap-size}[A] \leftarrow \text{heap-size}[A] + 1 \\
2 & i \leftarrow \text{heap-size}[A] \\
3 & \text{while } i > 1 \text{ and } A[\text{PARENT}(i)] < k \\
4 & \quad \text{do } A[i] \leftarrow A[\text{PARENT}(i)] \\
5 & \quad i \leftarrow \text{PARENT}(i) \\
6 & A[i] \leftarrow k \\
\end{align*}
\]

(a) What does the heap look like inserting keys from the sequence: \{10, 3, 1, 12, 20, 18, 14, 16\}? 

(b) What is the height of the heap from inserting keys from the sequence: \{10, 3, 1, 12, 20, 18, 14, 16\}? 

(c) Write a function that will take a heap (stored in an array called A) and return the maximum value.
4. Give asymptotic upper and lower bounds for $T(n)$ for the following two recurrences. Make your bounds as tight as possible, and justify your answers:
Assume that $T(n)$ is constant for $n \leq 2$:

(a) $T(n) = 5T(n/3) + 1$

(b) $T(n) = 10T(n - 2) + n$
5. Assume $A[1..n]$ is an array.

FIND-KEY(A,k)
1 for i <- 1 to n
2 do if $A[i] = k$
3 then return i

(a) What are tight bounds on the worst case order of growth? Justify your answer:

(b) What are tight bounds on the best case order of growth? Justify your answer:

(c) What are tight bounds on the average case order of growth, assuming that all numbers in $A$ are randomly drawn from the interval $[1, n]$? Justify your answer:
6. Suppose that we have an array of \( n \) objects to sort and that the key of each record has the value \( \{0, 1, \ldots, k\} \). Assume that \( k \) is much smaller than \( n \) (\( k = o(n) \)). Give a simple, linear-time algorithm for sorting the \( n \)-objects.

7. Suppose that we have an array of \( n \) objects to sort, and there are no conditions on the keys.

   (a) What is the lower bound on the worst case running time of a comparison sort of the array \( A \)?

   (b) Write a sorting algorithm that sorts a list with the worst case running time you stated above: