Homework Assignment #2

Assigned: Tuesday 09/22/2015; Due: Thursday 10/08/2015 by 11:59pm through Oncourse.

(total: 100 points)

Problem 1 (5 points) Prove or disprove the following equation

\[(n + a)^b = \Theta(n^b)\]

where \(a\) and \(b > 0\) are constants.

Problem 2 (6 points) Prove or disprove the following expressions

a) (2 points) \(2^{n+1} = O(2^n)\)
b) (2 points) \(2^{2n} = O(2^n)\)
c) (2 points) \(3^n = O(2^{2n})\)

Problem 3 (4 points) In the HIRE-ASSISTANT program from the textbook (Chapter 5), assuming that the candidates are presented in a random order, what is the probability that an exactly one hire will be made? Then, find the probability that exactly \(n\) hires will be made.

Problem 4 (5 points) In the HIRE-ASSISTANT program from the textbook (Chapter 5), assuming that the candidates are presented in a random order, what is the probability that exactly two hires will be made?

Problem 5 (5 points) Explain how to implement two stacks in one array \(A[1..n]\) in such a way that neither stack overflows unless the total number of elements in both stacks together is \(n\). The PUSH and POP operations should run in \(O(1)\) time.

Problem 6 (5 points) Propose a \(\Theta(n)\)-time nonrecursive procedure that reverses a singly linked list of \(n\) elements. The program should use no more than constant storage beyond that needed for the list itself.

Problem 7 (5 points) Draw a picture of the sequence 13, 4, 8, 19, 5, 11 stored as a doubly linked list using the multiple-array representation. Do the same for the single-array representation.
Problem 8 (10 points) Draw a binary tree rooted at index 6 that is represented by the following fields

<table>
<thead>
<tr>
<th>Index</th>
<th>Key</th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>8</td>
<td>Nil</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>10</td>
<td>Nil</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>Nil</td>
<td>Nil</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Nil</td>
<td>Nil</td>
</tr>
<tr>
<td>8</td>
<td>14</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td>Nil</td>
<td>Nil</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>Nil</td>
<td>Nil</td>
</tr>
</tbody>
</table>

Problem 9 (10 points) Demonstrate the insertion of the keys 5, 28, 19, 15, 20, 33, 21, 17, 10 into a hash table with collisions resolved by chaining. Let the table have 9 slots and let the hash function be \( h(x) = x \mod 9 \). Assume that buckets start with 0 and end with \( m - 1 \).

Problem 10 (15 points) Consider a version of the division method in which \( h(x) = x \mod m \), where \( m = 2^p - 1 \) and \( x \) is a character string interpreted in radix \( 2^p \). Show that if string \( s \) can be derived from string \( t \) by permuting its characters, then \( s \) and \( t \) hash to the same value. Assume that buckets start with 0 and end with \( m - 1 \).

Problem 11 (30 points) Data structures for storing sets of integer variables.

1. Stack operations. (15 points) Write Matlab functions that implement basic stack operations \( \text{PUSH} \) and \( \text{POP} \). Implement a stack of size \( n \) using the array representation, but make sure you support overflow and underflow cases (by using function \( \text{error} \)). In particular, the stack should consist of a record variable \( S \) that contains a counter head and array \( A \). For example, the 5th element of the array should be accessed as \( S.A(5) \) while the head should be accessed as \( S.head \). The functions should be:

   ```matlab
   function [] = push(x)
   function [x] = pop()
   ```

   Stack \( S \) should be a global variable (type “help global” in Matlab). It should be initialized in the main script file that you can use to test your code.

2. Singly-linked list operations. (15 points) Write Matlab functions to implement singly-linked list operations \( \text{insert}(x) \), \( \text{delete}(x) \) and \( \text{member}(x) \). The linked list should consist of two arrays (next and key), both under the structure \( L \), and containing \( n \) positions. The list should also manage garbage collection. For example, to access the 5th element in the array key, you would
use $L$.key(5), while to access 7th position of the array next you would use $L$.next(7). List $L$ should also contain variables head and free. Once the list is created, it should be seen as global in functions insert, delete and member. In particular, the functions should be initialized as:

$$[\text{indicator}] = \text{insert}(x),$$ where $x$ is an element and indicator is 0 if insertion was successful and 1 if the list is full;

$$[\text{indicator}] = \text{delete}(x).$$ Variable indicator should be returned as 0 if deletion was successful, 1 if the list is empty and 2 if an element could not be found.

$$[\text{indicator}] = \text{member}(x),$$ where $x$ is an element and indicator is 1 if it is contained in the list and 0 otherwise.

---

**Homework policies:**

This assignment is strictly individual. All code (if applicable) should be turned in when you submit your assignment (as hard copy).

Policy for late submission assignments: Unless there are legitimate circumstances, late assignments will be accepted up to 5 days after the due date and graded using the following rule:

<table>
<thead>
<tr>
<th>On Time</th>
<th>1 Day Late</th>
<th>2 Days Late</th>
<th>3 Days Late</th>
<th>4 Days Late</th>
<th>5 Days Late</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your Score $\times 1$</td>
<td>Your Score $\times 0.9$</td>
<td>Your Score $\times 0.7$</td>
<td>Your Score $\times 0.5$</td>
<td>Your Score $\times 0.3$</td>
<td>Your Score $\times 0.1$</td>
</tr>
</tbody>
</table>

For example, this means that if you submit 3 days late and get 80 points for your answers, your total number of points will be $80 \times 0.5 = 40$ points.

All the sources used for problem solution must be acknowledged, e.g. web sites, books, research papers, personal communication with people, etc. Academic honesty is taken seriously; for detailed information see Indiana University Code of Student Rights, Responsibilities, and Conduct.

-----

Good luck!