Midterm Exam Summer II 2003 KEY

I.

1. c
2. b
3. c
4. a
5. b
6. e
7. c
8. b
9. a
10. d

II.

1. T(n) = n – 1 + log n
2. T(n) is O(n)
   T(n) is \( \Omega(\log n) \)

III.

1. Let \( p \) be the size of a pointer in bytes: Array implementation uses \((3p)100=300p\) bytes, linked implementation uses \(60(3p + p) = 240p\) bytes. Linked list is thus more space efficient in this case.
2. Array more efficient when \( n(factor) < c(factor + 1) \)
   Linked list more efficient when \( n(factor) > c(factor + 1) \)
3. Revise code so that it checks if pos is closer to tail than to head. If so, start with \( fence = tail \) and use previous pointer to reach position. (Can also check if pos is closer to current value of fence).
4.

<table>
<thead>
<tr>
<th>Method</th>
<th>Array</th>
<th>Singly-Linked List</th>
<th>Double-Linked List</th>
</tr>
</thead>
<tbody>
<tr>
<td>setpos()</td>
<td>O(1)</td>
<td>O(n)</td>
<td>O(n)</td>
</tr>
<tr>
<td>insert()</td>
<td>O(n)</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>append()</td>
<td>O(1)</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>prev()</td>
<td>O(1)</td>
<td>O(n)</td>
<td>O(1)</td>
</tr>
</tbody>
</table>

IV.

1. Many answers (make sure all internal nodes have EXACTLY two children)
2. I have no time to draw this electronically, but the leaves are B, D, F, and H
3. 60, 52, 30, 10, 47, 28, 7, 5, 9
4. 16, 7, 3, 12, 50, 38, 90 (Other answers possible)