Algorithms (2006 Fall) Midterm

1. (25 points) For each of the following statements, determine whether it is correct or not. You should explain why for your answers (True or False).

   (1) \( n^{1/\lg n} = \Omega(4^{\lg n}) \). [F, \( n^{1/\lg n} = 2 = \Theta(1) \neq \Omega(4^{\lg n}) \)]

   (2) If \( f(n) \) does not belong to the set \( o(g(n)) \), then \( f(n) = \Omega(g(n)) \). [F, see counterexamples in the lecture note]

   (3) Sorting 6 elements with a comparison sort requires at least 15 comparisons in the worst case. (\( \lg 6 = 2.5 \)) [F, \( 2^9 < 6! < 2^{10} \), so at least 10 comparisons]

   (4) In a heap with size \( n \), the subtree-sizes of root’s children are at most \( 3n/4 \). [F, at most, see Ch6 slide p14–15]

   (5) A set of \( n \) integers in the range \{1, 2, \ldots, n\} can be sorted by RADIX-SORT in \( O(n) \) time by running QUICK-SORT on each bit of the binary representation. [F, QUICK-SORT is not a stable sorting algorithm or its running time is \( \Omega(n \lg n) \).]

2. (20 points) Give asymptotic upper and lower bounds for \( T(n) \) in each of the following recurrences by the assigned method listed in (). Assume that \( T(n) \) is constant for \( n \leq 2 \). Make your bounds as tight as possible.

   (1) \( T(n) = 2T(n/3) + n\lg n \)

   Sol: By case 3 of the Master Method, we have \( T(n) = \Theta(\lg n) \).

   (2) \( T(n) = T(2^{\lg n}) + 1 \)

   Sol:

   \[
   \text{Suppose } m = \lg n \Rightarrow n = 2^m \Rightarrow T(2^m) = T(2^{\sqrt m}) + 1
   
   \text{Let } S(m) = T(2^m) \Rightarrow S(m) = S(\sqrt m) + 1
   
   \text{Suppose } p = \lg m \Rightarrow m = 2^p \Rightarrow S(2^p) = S(2^{p/2}) + 1
   
   \text{Let } S'(p) = S(2^p) \Rightarrow S'(p) = S'(p/2) + 1 \Rightarrow S'(p) = O(p)
   
   S'(p) = S(2^p) = S(m) = T(2^m) = T(n) \Rightarrow T(n) = O(p) = O(p \lg m) = O(p \lg \lg n)
   \]

   (3) \( T(n) = 10T(n/3) + 17n^{1.2} \)

   Sol: Since \( \log_3 9 = 2 \), so \( \log_3 10 > 2 > 1.2 \). Case 1 of Master Method applies, \( \Theta(n^{\log_3 10}) \).

   (4) \( T(n) = \sqrt n T(\sqrt n) + 100n \)

   Sol: Master’s theorem doesn’t apply here directly. Pick \( S(n) = T(n)/n \). The recurrence becomes \( S(n) = S(\sqrt n) + 100 \). The solution of this recurrence is \( S(n) = \Theta(\lg \lg n) \). (You can do this by a recursion tree, or by substituting \( m = \lg n \) again.) Therefore, \( T(n) = \Theta(n \lg \lg n) \).

3. (10 points) Insert a sequence of keys \( (5 \ 8 \ 2 \ 3 \ 9 \ 4 \ 7 \ 10 \ 1 \ 6) \) into an empty heap (max-heap). Show the heap trees step by step.
4. (15 points) Please describe the optimal substructures for the COIN-CHANGE problem. Please also calculate how many different subproblems overall and how many choices we have in determining which subproblem(s) to use in an optimal solution. The COIN-CHANGE problem: An amount of money \( M \), and an array of \( d \) denominations \( c = (c_1, c_2, \ldots, c_d) \), in a decreasing order of value (\( c_1 > c_2 > \ldots > c_d \)). Please find a list of \( d \) integers \( i_1, i_2, \ldots, i_d \) such that \( c_1 \cdot i_1 + c_2 \cdot i_2 + \ldots + c_d \cdot i_d = M \) and \( i_1 + i_2 + \ldots + i_d \) is minimal.

Ans: (1) \#subproblems = \( O(M) \), \#choice to determining which subproblems = \( d \)
5. (15 points) For a matrix multiplication problem, \( M_{7\times3} M_{3\times6} M_{6\times2} M_{2\times8} M_{8\times4} \). Please derive the most efficient matrix multiplication method and also briefly describe your method and the optimal multiplication cost.

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
p & 0 & 1 & 2 & 3 & 4 & 5 \\
\hline
7 & 3 & 6 & 2 & 8 & 4 & \\
\hline
\end{array}
\]

\[
\begin{align*}
M[1,3] &= 78 \\
(k=1) \Rightarrow M[1,1] + M[2,3] + P[0]*P[1]*P[3] = 0 + 36 + 42 = 78 \text{ (min)} \\
(k=2) \Rightarrow M[1,2] + M[3,3] + P[0]*P[2]*P[3] = 126 + 0 + 84 = 210 \\
M[2,4] &= 84 \\
M[3,5] &= 112 \\
M[1,4] &= 190 \\
(k=1) \Rightarrow M[1,1] + M[2,4] + P[0]*P[1]*P[4] = 0 + 84 + 168 = 112 \\
(k=3) \Rightarrow M[1,3] + M[4,4] + P[0]*P[3]*P[4] = 78 + 0 + 112 = 190 \text{ (min)} \\
M[2,5] &= 124 \\
M[1,5] &= 198 \\
(k=1) \Rightarrow M[1,1] + M[2,5] + P[0]*P[1]*P[5] = 0 + 124 + 84 = 208 \\
(k=3) \Rightarrow M[1,3] + M[4,5] + P[0]*P[3]*P[5] = 78 + 64 + 56 = 198 \text{ (min)} \\
\end{align*}
\]
![Table Image]

6. (15 points) Please fill in the following blanks.

<table>
<thead>
<tr>
<th></th>
<th>Insertion Sort</th>
<th>Merge Sort</th>
<th>Heap Sort</th>
<th>Quick Sort</th>
</tr>
</thead>
<tbody>
<tr>
<td>sort in place (T/F)</td>
<td>(1) T</td>
<td>(5) F</td>
<td>(9) T(F)</td>
<td>(12) T</td>
</tr>
<tr>
<td>stable (T/F)</td>
<td>(2) T</td>
<td>(6) T</td>
<td>(10) F</td>
<td>(13) F</td>
</tr>
<tr>
<td>worst case time</td>
<td>(3) O(n²)</td>
<td>(7) Θ(nlg n)</td>
<td>(11) Θ(nlg n)</td>
<td>(14) O(n²)</td>
</tr>
<tr>
<td>best case time</td>
<td>(4) Θ(n)</td>
<td>(8) Θ(nlg n)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>average case time</td>
<td></td>
<td></td>
<td>(15) Θ(nlg n)</td>
<td></td>
</tr>
</tbody>
</table>

* You should consider the **tightest asymptotic notations** in the running time analysis.