CS 371
Homework Assignment 1

Given: February 03, 2015                  Due: February 17, 2015

Note: The homework is due on Tuesday, February 17 at the beginning of the class.

Please write concise and clear solutions; you will get only a partial credit for correct solutions that are either unnecessarily long or not clear.

You are allowed to discuss ideas for solving homework problems in groups of up to 3 people but you must write your solutions independently. Also, you must write on your homework the names of the people with whom you discussed.

Finally, you are not allowed to use any material outside of the class notes and the textbook. Any violation of this policy may seriously affect your grade in the class.

In this homework, log refers to log_2 and lg also refers to log_2.

1. (a) For each pair of expressions (A, B) in the table below, indicate whether A is O, o, Ω, ω or Θ of B. Assume that k ≥ 1, ϵ > 0, and c > 1 are constants. Your answer should be in the form of the table with “yes” or “no” written in each box. No justification is required.

   | A  | B  | O  | o  | Ω  | ω  | Θ  |
---|----|----|----|----|----|----|----|
| lg^k n | n^ε |    |    |    |    |    |
| n^k   | c^n |    |    |    |    |    |
| 2^n   | 2^{n/2} |    |    |    |    |    |
| n lg c | c^{lg n} |    |    |    |    |    |
| lg(n!) | lg(n^n) |    |    |    |    |    |

(b) Rank the following functions by order of their growth; i.e., find an arrangement g_1, g_2, ... of the functions satisfying g_1 = Ω(g_2), g_2 = Ω(g_3), .... No justification is required.

\[
\begin{align*}
(\sqrt{2})^{\lg n} & \quad n^2 \quad n! \quad \lg(n)!
\frac{3}{2} & \quad n^3 \quad \lg^2 n \quad \lg(n!)
2^{2^n} & \quad n^{1/\lg n} \quad \ln n \quad n \cdot 2^n
n \lg \lg n & \quad \ln n \quad 1 \quad 2^{\lg n}
(\lg n)^{\lg n} & \quad e^n \quad 4^{\lg n} \quad (n + 1)!
\sqrt{\lg n} & \quad 2\sqrt{\lg n} \quad n \quad 2^n
n \lg n & \quad 2^{n+1} \quad 2^{100^{100}}
\end{align*}
\]
2. Prove or disprove the following. In case of a proof, use the definitions of $O$, $\Omega$, $\Theta$ and give values of the constants in the definitions for which the conditions in the definition hold.

(i) $5n\sqrt{n} = O\left(\frac{1}{7}n^2 - 10\right)$
(ii) $2^{5\lg n + \lg \lg n} \lg(n^5) = O(4^{3\lg n})$

3. Let $f(n)$ and $g(n)$ be two functions from the set of positive integers to the set of positive integers. Prove or disprove the following.

(i) There exist functions $f(n)$ and $g(n)$ such that neither $f(n)$ is $O(g(n))$ nor $g(n)$ is $O(f(n))$.
(ii) If $f(n) = O(s(n))$ and $g(n) = O(r(n))$ then $f(n)/g(n) = O(s(n)/r(n))$.

4. Consider the following code fragment.

\begin{verbatim}
for i = 1 to n + 100 do
    for j = 1 to i * n do
        sum = sum + j
    for k = 1 to 3n do
        c[k] = c[k] + sum
\end{verbatim}

a. For the above code fragment give a bound of the form $O(f(n))$ on its running time on an input of size $n$. Justify your answer.

b. For this same function $f$, show that the running time of the algorithm on an input of size $n$ is also $\Omega(f(n))$. Justify your answer. (This shows an asymptotically tight bound of $\Theta(f(n))$ on the running time.)

5. Consider the following code fragment.

\begin{verbatim}
sum = 0;
for (int i=1; i <= n; i++)
    for (int j=1; j <= i; j*=2)
        for (int k=1; k <= j; k*=3)
            sum++;
\end{verbatim}

a. For the above code fragment give a bound of the form $O(f(n))$ on its running time on an input of size $n$. Justify your answer.

b. For this same function $f$, show that the running time of the algorithm on an input of size $n$ is also $\Omega(f(n))$. Justify your answer. (This shows an asymptotically tight bound of $\Theta(f(n))$ on the running time.)

A[i + 1] + ⋯ + A[j]. (The value of the array entry B[i, j] is left unspecified whenever i ≥ j, so it doesn’t matter what is the output of these values.)

Here is a simple algorithm to solve this problem.

```plaintext
for (i=1; i <= n; i++)
    for (j=i+1; j <= n; j++)
        Add up array entries A[i] through A[j]
        Store the result in B[i,j]
```

a. For the above code fragment give a bound of the form $O(f(n))$ on its running time on an input of size $n$. Justify your answer.

b. For this same function $f$, show that the running time of the algorithm on an input of size $n$ is also $\Omega(f(n))$. (This shows an asymptotically tight bound of $\Theta(f(n))$ on the running time.)

c. Although the algorithm you analyzed in parts (a) and (b) is the most natural way to solve the problem – after all, it just iterates through the relevant entries of the array $B$, filling in a value for each – it contains some highly unnecessary sources of inefficiency. Give a different algorithm to solve this problem, with an asymptotically better running time. In other words, you should design an algorithm with running time $O(g(n))$, where $\lim_{n \to \infty} g(n)/f(n) = 0$.

7. Decide whether the following statements are true or false. If it is true, argue its correctness, otherwise, give a counterexample.

   Define a couple $(m, w)$ to be nice if $w$ is ranked first on $m$’s preference list and $m$ is ranked first on $w$’s preference list.

   a. For every instance of the Stable Marriage Problem there is a stable matching containing some nice couple $(m, w)$.
   
   b. Consider an instance of the Stable Matching Problem in which there exists a nice couple $(m, w)$. Then in every stable matching $S$ for this instance, the pair $(m, w)$ belongs to $S$.
   
   c. Consider an instance of the Stable Marriage Problem in which there are $n$ disjoint nice couples. Then there is a unique stable matching for this instance.
   
   d. Consider an instance $I$ of the Stable Marriage Problem for which there is a unique stable matching. Then $I$ must consist of $n$ disjoint nice couples.

8. The Hotel Partner Problem is as follows. There are $2n$ people and $n$ hotel rooms. Each person maintains a preference list of the remaining $2n - 1$ people. The objective is to assign two people to each room such that the assignment is stable. An assignment is stable if there are no two people assigned to different rooms who prefer each other over their current partners. Give an instance of the Hotel Partner Problem for which there is no stable solution.
9. Chapter 1, Problem 6 (page 25). Solve this problem by reducing it to the Stable matching problem.

10. Consider an instance of the stable matching problem with $n$ men and $n$ women. Let $X$ and $Y$ be some two stable matchings for this instance. We now construct a new pairing $Z$ as follows. For each man $m$, if $X$ pairs him with a woman $w^m_x$ and $Y$ pairs him with a woman $w^m_y$ then in $Z$ the man is paired with the woman he prefers most among $w^m_x$ and $w^m_y$. Note that $w^m_x$ and $w^m_y$ could be the same woman. Prove or disprove that $Z$ is a stable matching.

Now consider a pairing $Z'$ in which a man $m$ is paired with the woman he prefers the least among $w^m_x$ and $w^m_y$. Prove or disprove that $Z'$ is a stable matching.