Homework 1
Fundamental Algorithms, Spring 2012, Professor Davis

Assigned: Jan. 23
Due: Jan 30, in class.

INSTRUCTIONS:

• Remember that we have a “no late homework” policy. Special permission must be obtained in advance if you have a valid reason.

• Please read questions carefully. When in doubt, please ask.

• Bring your questions to the recitation session.

• Please read our homework policy and know the department’s academic integrity policy. You may work in collaboration, but regardless, you must write up all solutions yourself.

Solutions to this problem set will be graded by Rameez Loladia rgl251@nyu.edu. Your solutions should either be submitted in class or emailed to him.

In what follows log(·) denotes the logarithm function with base 2.

Problem 1 (20 points)

Let $A[1, \ldots, n]$ be an array of $n$ distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair $(i, j)$ is called an inversion of $A$.

(a) List the five inversions of the array $\langle 2, 3, 8, 6, 1 \rangle$.
(b) What array with elements from the set $\{1, 2, \ldots, n\}$ has the largest number of inversions? State this bound.
(c) What is the relationship between the running time of INSERTION_SORT and the number of inversions in the input array? Justify your answer.

Problem 2 (20 points)

For each of the following pairs of functions $f(n)$ and $g(n)$, state whether $f$ is $O(g)$; whether $f$ is $o(g)$; whether $f$ is $\Theta(g)$; whether $f$ is $\Omega(g)$; and whether $f$ is $\omega(g)$. (More than one of these can be true for a single pair.)

A. $f(n) = 3n^2 + 2; g(n) = n^2 + 5n$
B. $f(n) = \log(n^2); g(n) = \log(n^3)$.
C. $f(n) = \log(2^n); g(n) = \log(n^2)$.
D. $f(n) = n^3 \cdot 2^n; g(n) = n^2 \cdot 3^n$.
E. $f(n) = (n^n)^2; g(n) = n^{(n^2)}$.
Problem 3 (20 points)

List the following functions in increasing order of growth. If two functions have the same order of growth, state the fact.

\[
\begin{align*}
&n, & n \cdot (\log(n))^2, & n/\log(n), & n \cdot \log(n^2), & n^2 \cdot \log(n), \\
&2^{\log(n)}, & n^22^n, & 3^n/n, & \log(\log(n)), & (\log(n))^2, \\
&\log(n^2), & \log(n)\cdot \log(\log(n)), & (n!)^2, & (n^2)!.
\end{align*}
\]

Problem 4 (20 points)

The following two functions both compute the same thing: They take as arguments two arrays \(A\) and \(B\) and return TRUE if every element of \(A\) is less than every element of \(B\).

```c
bool AllLessThan1(int[] A, B) {
    for (i = 1 to |A|)
        for (j = 1 to |B|)
            if (A[i] >= B[j]) return FALSE;
    return TRUE;
}
```

```c
bool AllLessThan2(int[] A, B) {
    largeA = A[1]
    for (i = 2 to |A|)
        if (A[i] > largeA) largeA = A[i];
    for (j = 1 to |B|)
        if (largeA >= B[j]) return FALSE;
    return TRUE;
}
```

A. Give the asymptotic worst-case running time of each algorithm as a function of \(|A|\) and \(|B|\) (the lengths of \(A\) and \(B\).) When does the worst case occur?

B. Give the asymptotic best-case running time of each algorithm. When does the best case occur?