

**Due:** Wed., 6 October 2004

**Homework Exercises.** Create a directory to hold your answers to this homework set. Copy the files from `$master/hw/hw5` into this directory. Put non-program answers into a file `hw5.txt` (plain text, please, no PDF, RTF, or Word files). Use the command `submit hw5` to submit your solutions to the problems below.

1. Exercise 1.1 from the *Data Structures (Into Java)* reader.
2. Exercise 1.2 from the *Data Structures (Into Java)* reader.
3. Suppose that  $f(n)$  is a positive, non-decreasing function. Show that  $\lceil f(n) \rceil \in O(f(n))$ .
4. Suppose that  $p(x)$  is any polynomial in  $x$  with positive coefficients. Show that  $\log p(x) \in O(\log x)$ .
5. Show that  $\log_b f(x) \in \Theta(\log f(x))$  for any constant  $b > 1$ .
6. What is the worst-case running time for the following program fragment?

```
int j;
j = 0;
for (int i = 0; i < N; i += 1) {
    for ( ; j < M; j += 1) {
        if (bump (i, j))
            break;
    }
}
```

Assume that `M` and `N` are integer constants, and `bump` is a constant-time ( $O(1)$ ) method that returns a boolean result. We're looking for a  $\Theta$  result that uses `M` and `N`.

7. [Goodrich & Tamassia] Suppose that  $A$  is an  $n \times n$  array of 1's and 0's with the property that all the 1's in a row come before all the 0's in that row. The array is actually huge ( $n > 500000$ ), and instead of actually being stored as a Java array, it is represented by a `BitMatrix` object with a method `.get(i, j)`, which returns  $A_{ij}$  ( $i$  is the row,  $j$  the column). Fill in the method `mostOnes(A)` in the template file `BigMat.java` so that it returns the index of the row of  $A$  that contains the most 1's. When several rows contain the largest number of 1's, return the smaller index. Your method must operate in  $O(n)$  time (*not*  $O(n^2)$  time). Your program will be given a time limit that requires it to operate in better than  $O(n^2)$  time.