

ECE 1320 Optimization Methods
Winter 2003

Homework 2: Due in class Thursday January 23 2003

- This test contains 6 problems. They allow you to earn 100 points.
- Show your work, as partial credit can be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. **Be neat.**
- **No late submissions will be accepted.**
- Only homework returned in a 9in \times 12in envelope will be accepted. (If you cannot find such envelope, ask the Instructor.) Please, write your name and the class name (ECE 1320) on the envelope (write clearly, please).

Write your name here: _____

- **Problem # 1 [10 points]**. Consider the function *piccolo* from homework 1. **(a)** Discuss the worst case time complexity of your solution. **(b)** Can a function that solves this problem be written that requires less than n comparisons? (n is the number of the elements in the array.)

- **Problem # 2 [10 points]**. Discuss the time complexity of your function *printSingles* from homework 1 in the case when the element which are input are not just 20 but a generic number $n \geq 1$. Can you obtain an *optimal algorithm*?

- **Problem # 3 [20 points].** Prove or disprove the following:

$$\sum_{i=1}^n i^h \in O(n^{h+1}), h \geq 0.$$

- **Problem # 4 [20 points].** Prove that $\lceil \log n \rceil \in O(n)$.

- **Problem # 5 [20 points]**. What is the value returned by the following function? Express your answer as a function of n . Give, using the “big O” notation, the worst-case running time.

```
int mystery( int n ) {
    int r = 0;
    for( int i = 1; i < n; i++ )
        for( int j = i + 1; j < n + 1; j++ )
            for( int k = 1; k < j + 1; k++ )
                r++;
    return r;
}
```

- **Problem # 6 [20 points]**. Hardware vendor XYZ Corp. claims that their latest computer will run 100 times faster than that of their competitors, Prunes, Inc. If the Prunes, Inc. computer can execute a program on input size n in an hour, what size input can XYZ's computer execute in one hour for each algorithm with the following growth rate equations?

1. n
2. n^2
3. n^3
4. 2^n

- **Bonus problem [20 points]**. Consider the following, fundamental question: “Given a number $n > 1$, is n prime, i.e., is n divisible only by 1 and itself?”

The question is answered by the following function (where the C++ operator `%` is the modulus operator, which returns the remainder of the integer division between its operands):

```
bool prime( int n ) {
    int j = 2;
    while ( j < n )
        if ( n % j == 0 )
            return false;
        else
            j++;
    return true;
}
```

Function `prime` attempts to divide n by every number j in the range $2, \dots, n - 1$ and returns *true* only if no number j that divides n has been found.

Is this function time complexity polynomial in the *size of the input*? Justify your answer.