

HOMEWORK 2  
Due Friday, September 25, 2009 at the beginning of class

For general homework policies and our suggestions, please see the policy document.

For this homework, you can assume that addition, subtraction, multiplication and division of two numbers can be done in  $O(1)$  time. (Side question: Is this assumption justified?)

Do not turn the first problem in and no collaboration is allowed on the second problem.

1. (**Do NOT turn this problem in**) This problem is just to get you thinking about asymptotic analysis and input sizes.

An integer  $n \geq 2$  is a prime, if the only divisors it has is 1 and  $n$ . Consider the following algorithm to check if the given number  $n$  is prime or not:

For every integer  $2 \leq i \leq \sqrt{n}$ , check if  $i$  divides  $n$ . If so declare  $n$  to be *not* a prime.

If no such  $i$  exists, declare  $n$  to be a prime.

What is the function  $f(n)$  such that the algorithm above has running time  $\Theta(f(n))$ ? Is this a polynomial running time— justify your answer. (A tangential question: Why is the algorithm correct?)

2. (**You must work on this problem on your own: NO collaboration is allowed**) (20 points) Order the following running times in ascending order so that if one (let's call it  $f(n)$ ) comes before another (let's call it  $g(n)$ ), then  $f(n)$  is  $O(g(n))$ :

$$2^{n^2}, 2^{\log_2 n}, n^2, n!, 2^{(\log_3 n)^2}, 2^n.$$

Briefly argue why your order is correct (formal proof is not needed).

(For this problem, it might help to know Stirling's approximation:

$$n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n e^{\lambda_n},$$

where

$$\frac{1}{12n+1} < \lambda_n < \frac{1}{12n}.)$$

3. (25 + 25 + 10 = 50 points) Exercise 6 in Chapter 2.  
4. (20 points) Exercise 6 in Chapter 1.

(In real life, you will almost never come across a problem whose description will match exactly with one you will see in this course. More often, you will come across problems that you have seen earlier *but* are stated in a way that don't look like the version you have seen earlier. This problem is trying to simulate that situation. In algorithms-speak, you will have to *reduce* the problem here to one that you have seen already.)