

CS 477/677 Analysis of Algorithms

Homework 1

Due September 9, 2008

Note: Students in the CS 477 section must solve problems 1 through 3 and can solve either problem 4 or 5 for extra credit. Students in the 677 section must solve problems 1 through 4 and can solve problem 5 for extra credit.

1. [30 points] Arrange the following list of functions in ascending order of growth rate. That is, if function $g(n)$ immediately follows function $f(n)$ in your list, then $f(n)$ should be $O(g(n))$.

$$f_1(n) = n^{2.5}$$

$$f_2(n) = \sqrt{2n}$$

$$f_3(n) = n + 10$$

$$f_4(n) = 10^n$$

$$f_5(n) = 100^n$$

$$f_6(n) = n^2 \log n$$

2. [35 points] Using mathematical induction, show that the following relations are true for every $n \geq 1$:

$$\text{a) } \sum_{i=1}^n (-1)^{i+1} i^2 = \frac{(-1)^{n+1} n(n+1)}{2}$$

$$\text{b) } \sum_{i=1}^n \frac{1}{(2i-1)(2i+1)} = \frac{n}{2n+1}$$

3. [35 points] Write the recurrence (without solving it) for the running time of the following algorithm:

Algorithm: example(n)

```
{
  if (n = 1)
    return
  for i = 1 to n
    x = x + 1
  y = example(n-1) + example(n-2)
}
```

4. [Graduate only - 20 points] Assume you have functions f and g such that $f(n)$ is $O(g(n))$. For each of the following statements, decide whether you think it is true or false and give a proof or a counter-example.

- a) $2^{f(n)}$ is $O(2^{g(n)})$
- b) $f(n)^2$ is $O(g(n)^2)$

5. [Extra credit - 20 points] Assume you have five algorithms with the running times listed below (these are the exact running times). How much slower do each of these algorithms get when you (a) double the input size, or (b) increase the input size by 1?

- i) n^2
- ii) n^3
- iii) $100n^2$
- iv) $n \lg n$
- v) 2^n