Assignment 1, CISC 365, Fall 2010

Due Friday, September 24 at the 10:30 AM lecture

D. Blostein

Handwritten answers are fine, but please write neatly. Do not use a calculator for problems 1 to 7. If you have trouble figuring out how to solve problems 1 to 7 without a calculator, see a TA or Prof. Blostein for help.

Background Information for Assignment 1

Math notation is used heavily algorithm analysis. This assignment reviews common notation, including Σ, Π, log, lim, ∀, and ∃. If this assignment is easy for you, great! Otherwise, take the time to complete the assignment carefully, so that you are well-equipped to understand the textbook and lectures. We begin with a page and a half of background information, followed by the assignment problems.

Free Variables and Bound Variables

In this course, symbols such as *i, k, x, y* are integers, unless noted otherwise. Each symbol occurs as a *free variable* (it can take on any integer value), or as a *bound variable* (it only takes on the values given within the expression). The use of free variables is a shorthand notation that omits ∀.

For example, the equality in problem 2(a) of this assignment is $\sum_{i=1}^{n}(x+y) = \left(\sum_{i=1}^{n} x\right) + \left(\sum_{i=1}^{n} y\right)$

This equality has three free variables (n, x, y) and one bound variable (i). It can be written more explicitly as

$$\forall x\ \forall y\ \forall n\ \sum_{i=1}^{n}(x+y) = \left(\sum_{i=1}^{n} x\right) + \left(\sum_{i=1}^{n} y\right)$$

Proving and Disproving Equalities that have Free Variables

To prove an equality, it is necessary to prove that the equality holds no matter what values are chosen for the free variables. To disprove an equality, it is sufficient to show *one* set of values for the free variables that makes the equality false.

For example, to prove the above equality, you must show that the equality holds for *any* integer values *n, x, y*. To disprove Equality 2(a), you need to give one counterexample: find a particular set of values for *n, x, y* which make the equality false. (Notice that the variable *i* is not mentioned; this is because *i* is a bound variable.)

Notation for products

Product notation is used in problems 1 and 2. The symbol Π (which is the capital "pi" from the Greek alphabet) is used for products, analogous to the use of Σ (capital sigma) for sums. For example, the inequality

$$10{,}000! < (1{,}000^{1000}) * (2000^{1000}) * \ldots * (10{,}000^{1000})$$

can be written as

$$10{,}000! < \prod_{i=1}^{10} (1000\,i)^{1000}$$

Sum and product notation can be used together, to denote a sum of products or a product of sums. The notation in problem 1(c) shows a sum of sums: $\sum_{k=1}^{2}\sum_{i=1}^{k} 1$. This could be evaluated using nested *for* loops:

```
sum = 0
for k = 1 to 2
    for i = 1 to k
       sum = sum + 1
```

Review of logarithms

Logarithms are used in problem 6 of this assignment. You will see them often in algorithm analysis, for example, when a sorting algorithm is characterized as running in O(n log n) time.

The base-10 logarithm of a real number K>0.0 is written as $\log_{10} K$.

Saying that $x = \log_{10} K$

is the same as saying $10^x = K$.

If K<1.0 the logarithm is negative. If K>1.0, the logarithm is positive. Here is a useful approximation for the logarithm of K, when K is an integer:

$\log_{10} K \approx$ number of digits needed to write K in base 10. (To be exact, $\lceil \log_{10}(K+1) \rceil$ digits are needed.)

$\log_2 K \approx$ number of digits needed to write K in base 2. Other bases are similar.

For example, $\log_{10} 83{,}223{,}189 \approx 8$. Many formulas involving logarithms are easy to understand and remember using this approximation. For example, the formula "$\log_{10} (10*K) = (\log_{10} K) + 1$" says that in base 10 it takes one more digit to write down the integer $10*K$ than to write down the integer K.

Assignment 1 Problems

1. Evaluate the following sums and products. Some answers (such as (d)) are functions of the free variable n. In (b) and (c), the value inside the sigma is 1 (one), not i. The double sigma in (c) is analogous to nested "for" loops.

(a) $\sum_{i=1}^{4} i$ (b) $\sum_{i=1}^{3} 1$ (c) $\sum_{k=1}^{2} \sum_{i=1}^{k} 1$ (d) $\sum_{i=1}^{n} n$ (e) $\sum_{i=1}^{n} i$ (f) $\prod_{i=1}^{3} 2$ (g) $\prod_{i=-1}^{1} i$ (h) $\prod_{i=-87}^{43} i$ (i) $\prod_{i=1}^{4} i$ (j) $\prod_{i=1}^{4} n$

Hint for (e): The first term and last term sum to $1 + n$. The second term and second-to-last term sum to $2 + (n - 1)$. Continue pairing off terms which sum to n + 1. Figure out how many pairs of terms there are. You might find it easier to think separately about the cases "n is even" and "n is odd".

2. Are equalities (a) to (d) correct or incorrect? If you think the equality is correct, write an informal justification. (A proof is not required.) If you think the equality is incorrect, prove this by giving a counterexample.

(a) $\sum_{i=1}^{n} (x+y) = \left(\sum_{i=1}^{n} x\right) + \left(\sum_{i=1}^{n} y\right)$ (b) $\sum_{i=1}^{n} (x*y) = \left(\sum_{i=1}^{n} x\right) * \left(\sum_{i=1}^{n} y\right)$

(c) $\prod_{i=1}^{n} (x+y) = \left(\prod_{i=1}^{n} x\right) + \left(\prod_{i=1}^{n} y\right)$ (d) $\prod_{i=1}^{n} (x*y) = \left(\prod_{i=1}^{n} x\right) * \left(\prod_{i=1}^{n} y\right)$

3. Evaluate these floor and ceiling expressions. The floor of x is the largest integer less than or equal to x. The ceiling of x is the smallest integer greater than or equal to x.

(a) $\lceil 57.1 \rceil$ (b) $\lfloor 57.1 \rfloor$ (c) $\lceil 2\pi \rceil$ (d) $\lfloor 2\pi \rfloor$ (e) $\lceil 0.0 \rceil$ (f) $\lfloor 0.0 \rfloor$

4. Evaluate the following limits.

(a) $\lim_{n\to\infty}\left(3+\frac{1}{n}\right)$ (b) $\lim_{n\to\infty}\left(3+\log_{10}n\right)$ (c) $\lim_{n\to\infty}\frac{5n^2+10n+3}{7n^2}$ (d) $\lim_{n\to\infty}\frac{n!}{n^n}$

Hint: For (d): $\frac{n!}{n^n}=\frac{n}{n}*\frac{n-1}{n}*\frac{n-2}{n}*\ldots*\frac{1}{n}$. What is the limit of the ($1/n$) factor? How big are the other factors?

5. Which of the following assertions are true? Briefly justify your answers. Give a counterexample for the false assertions. [A counterexample for (a) consists of a particular value of x for which there is *no* value of y that makes the assertion true. A counterexample for (b) consists of showing that no matter what value is chosen for x, there is always a way to pick y to make the assertion false; y can be a function of x.]

(a) $\forall x\, \exists y\; y > x$ (b) $\exists x\, \forall y\; y \le x$ (c) $\exists x\, \exists y\; y > x$ (d) $\forall x\, \forall y\; y > x$

6. Estimate how big $\log_{10} K$ is, compared to $\log_2 K$.

Don't look up the "right answer" in some textbook. Think the problem through, particularly if you feel uncomfortable around logarithms. The goal is to get an intuitive understanding of logarithms, not to memorize formulas about them. To get started:

- Restate the problem in words: "It takes x digits to write the value K in base 10. How many digits does it take in base 2?" The answer should be some function of x. For example, maybe it takes $M + x$ digits or $M * x$ digits or $M * x^2$ digits. Your job is to figure out which function to use, and to estimate the value of M.
- Try out some examples. For example, $64_{10} = 1000000_2$. (The base of the number is written as a subscript.) Clearly, more digits are needed in base 2 than in base 10. How many more?
- Think about other bases that might be easier to compare to base 2. How many digits are needed to write K in **base 8 versus base 2**? (This is comparing $\log_8 K$ to $\log_2 K$.) How many digits are needed to write K in **base 16 versus base 2**? (This is comparing $\log_{16} K$ to $\log_2 K$.) The number of digits needed for base 10 is somewhere between the digits needed for base 8 and base 16...

7. Put these numbers into order from biggest to smallest. **Do not use a calculator**. One way to do this is by getting (rough) estimates of how many digits each number has (in base 10).

10^{200} 100030 500! $\log_{10} 238761$ sqrt(238761)

8. Lower bounds and upper bounds are used to characterize algorithm performance. Here is a review question on lower bounds and upper bounds. The problem arose in CISC324 (operating systems), where we were trying to estimate how many different ways the execution of one process can interleave with the execution of another process. (The operating system switches execution from one process to another at unpredictable times.) Here is an expression we wrote down[1], for two processes that each execute 10,000 statements: $\frac{10{,}001^{10{,}000}}{10{,}000!}$.

How do we get a rough idea of the size of this expression? The terms $10{,}001^{10{,}000}$ and 10,000! are too big to figure out on a calculator. Find a **lower bound** and an **upper bound** for this expression. A trivial lower bound is $\frac{10{,}001^{10{,}000}}{10{,}000!} \geq 0$. A trivial upper bound is $\frac{10{,}001^{10{,}000}}{10{,}000!} < 10{,}001^{10{,}000} < 100{,}000^{10{,}000} = \left(10^5\right)^{10{,}000} = 10^{50{,}000}$. Improve on this: find a lower bound greater than 0 and an upper bound less than $10^{50{,}000}$. You may use a calculator for this question. You are **not** expected to find an exact value. Just try to improve on the given bounds, reducing the distance between lower and upper bound. Both the lower bound and upper bound should be of the form 10^K, where K is an integer.

[1]You can skip this footnote if you don't care about how to count the number of interleaved executions of two processes. If you do care, you may realize that the expression in problem 8 does not count the number of interleaved executions correctly. We fixed this by realizing that the problem can be restated as "how many ways are there to merge two lists that each contain 10,000 items". The answer is $\frac{20{,}000!}{10{,}000! * 10{,}000!} = \binom{20{,}000}{10{,}000}$. The notation using the big parentheses is read aloud as "Twenty-thousand choose ten-thousand". For this CISC365 assignment, I want you to practice finding bounds for a large constant. I stuck with the first expression (the one in problem 8) because it is easier to find bounds for that expression than for the expression in this footnote.