CSCI 3100: Homework #1

Kate Holdener
kate.holdener@slu.edu

Saint Louis University — September 5, 2018

Introduction

The focus and purpose of this assignment is to refresh your skills in proofs, using the material we covered up to this point. Your solutions can be hand written or typed up. If your solution is hand written, it needs to be easily readable (if I can't read it, I can't grade it).

If you would like to type up your solutions, a good tool for that is LaTeX (which is what I used to type up this homework description). LaTeX is also something you'll most likely need to learn if you are planning to go to graduate school and there are plenty of LaTeX tutorials and templates available online. If you'd like to use LaTeX and don't know how to get started, ask me.

Homework solutions will be collected at the start of class on the day when they are due.

Problems

1. (3 points) Prove that running time $T(n) = 3n^2 + 5n + 2$ is $\Theta(n^2)$.

2. (2 points) Chapter "VIII Appendix: Mathematical Background" of your book contains much useful information. Read part "A: Summations" of that chapter for a refresher on variouls summation properties. Then show that $\sum_{k=1}^{n} \Omega f_k(i) = \Omega \sum_{k=1}^{n} f_k(i)$, using the linearity property of summations.

3. (5 points) Sort the following functions from asymptotically smallest to asymptotically largest, indicating ties, when appropriate. Use symbol $\equiv$ if two functions are asymptotically equivallent. This means that if $f(n) = \Theta(g(n))$, then you would write $f(n) \equiv g(n)$. Use symbol $\ll$ if one function is asymptotically larger than the other. This means that if $f(n) = O(g(n))$ and $f(n) \neq \Theta(g(n))$, you would write $f(n) \ll g(n)$. For example, functions $1, 2, n$ should be sorted as $1 \equiv 2 \ll n$.

1	n^3	$n^3 + n^2$	$nlg(n)$
$lg\sqrt{n}$	$lg(n)$	$lg(2^n)$	$n^{lg(n)}$
$\sum_{i=1}^{n+1} i$	$\sqrt{2^l g(n)}$	$nlg(n) + n^2$	$lg\sqrt{2^n}$

4. (5 points) Consider a simple for-loop below. Prove that the run time of this algorithm is $\Theta(n * g(n))$.

Algorithm 1: For-loop with $i + 1$ counter

for $i = 1; i \leq n; i = i + 1$ **do**
| Some operations with total run time $f(n) = \Theta(g(n))$
end

5. (5 points) Consider another for-loop below, noting that the counter i is multiplied by 2 at each iteration of the loop. Prove that the run time of this algorithm is $O(lg(n) * g(n))$.

Algorithm 2: For loop with $i * 2$ counter

for $i = 1; i \leq n; i = i * 2$ **do**
| Some operations with total run time $f(n) = \Theta(g(n))$
end