CIS 2910 Lab 5 – Solution and Discussion

Instructor: Dr. Charlie Obimbo (cobimbo@uoguelph.ca)
TAs: William Darling (wdarling@uoguelph.ca) and Eddie Ma (ema@uoguelph.ca)

Hello Class!
I am including a solution to the strong induction example that was shown during Lab 5. Also included solutions to problems on Assignment 3 that we discussed.

1 Strong Induction Example

Given a sequence g_0, g_1, g_2 ..., which we define *recursively* as:

$$\begin{aligned} g_0 &= 12 \\ g_1 &= 29 \\ g_n &= 5g_{n-1} - 6g_{n-2} \text{ for all integers } k \geq 2 \end{aligned}$$

Prove by strong induction that – *constant time*:

$$g_n = 5(3^n) + 7(2^n)$$

[Theorem]

$$\textbf{LHS: } g_n = \begin{cases} 12 & n = 0 \\ 29 & n = 1 \\ 5g_{n-1} - 6g_{n-2} & n \geq 2 \end{cases}$$

$$\textbf{RHS: } g_n = 5(3^n) + 7(2^n)$$

[Base Case] *two cases* – g_0, g_1

$$\begin{aligned} \textbf{n = 0: } g_0 &= 5(3^0) + 7(2^0) \\ g_0 &= 12 \\ \therefore \textbf{LHS} &= \textbf{RHS} \\ \textbf{n = 1: } g_1 &= 5(3^1) + 7(2^1) \\ g_0 &= 29 \\ \therefore \textbf{LHS} &= \textbf{RHS} \end{aligned}$$

[Proof by Strong Mathematical Induction]
[Inductive Hypothesis]

$$n \in \{0, ..., k\}$$
$$5g_{n-1} - 6g_{n-2} = 5(3^n) + 7(2^n) \text{ for all integers } k \geq 2$$

[Inductive Step] – Show that this works for $n = k + 1$

Starting with **LHS:**

$$\begin{aligned}
g_{k+1} &= 5g_k - 6g_{k-1} \\
&= 5[5(3^k) + 7(2^k)] - 6[5(3^{k-1}) + 7(2^{k-1})] \\
&= 5[5(3^k)] - 6[5(3^{k-1})] + 5[7(2^k)] - 6[7(2^{k-1})] \\
&= 5[5(3)(3^{k-1})] - 6[5(3^{k-1})] + 5[7(2)(2^{k-1})] - 6[7(2^{k-1})] \\
&= (3^{k-1})[5(5)(3) - 6(5)] + (2^{k-1})[5(7)(2) - 6(7)] \\
&= [75 - 30](3^{k-1}) + [70 - 42](2^{k-1}) \\
&= 45(3^{k-1}) + 28(2^{k-1}) \\
&= 5(3^2)(3^{k-1}) + 7(2^2)(2^{k-1}) \\
&= 5(3^{k+1}) + 7(2^{k+1})
\end{aligned}$$

We expect **RHS:**

$$g_{k+1} = 5(3^{k+1}) + 7(2^{k+1})$$ **QED**

2 Lab 5 Discussion from Assignment 3

2.1 Question 6b

An algorithm that runs in $\theta(n \lg n)$ solves a problem size of 1000 in 14 seconds. How long would it take to solve a problem of size 10000 in minutes?
Let's approximate $\lg 1000 \approx \lg 1024 \approx 10$ and also $\lg 10 \approx \lg 8 \approx 3$.

$$\begin{aligned}
\frac{1000(\lg 1000)}{10000(\lg 10000)} &= \frac{14}{60x} \\
\frac{\lg 1000}{10(\lg 10000)} &= \frac{14}{60x} \\
\frac{10}{10(\lg 1000 + \lg 10)} &= \frac{14}{60x} \\
\frac{60}{(10+3)} &\approx \frac{14}{x} \\
\frac{14(13)}{60} &\approx x \\
x &\approx 3
\end{aligned}$$

2.2 Question 7

Here's an example for finding $n \lg n$ using a spread sheet. You can do this in a couple of ways – one is to list all of the values between a lower bound (n) and an upper bound (n^2) – since $n \lg n$ can be found between the two. I chose bisection method. The left most column is used to find x in $(10^x)\lg(10^x)$. The value for x is changed in ever decreasing steps toward the solution. The centre column is an intermediate value $n = 10^x$. The right most column ($n \lg n$) is the output of the equation, and should approach the total number of operations expected.

Table 1: Solve for n in $n \lg n$ given $1E8$ total operations (1 second).

$(10^x)\lg(10^x)$	n	$n \lg n$
4	10000	132877.123795494 – linear (lower bound)
8	100000000	2657542475.90989 – quadratic (upper bound)
6	1000000	19931568.5693242
7	10000000	232534966.642115
6.5	3162277.66016838	68281583.5204621
6.75	5623413.25190349	126093877.676956
6.625	4216965.03428582	92806011.8744052
6.6875	4869675.25165863	108181755.022905
6.65625	4531583.63760082	100200491.719269
6.640625	4371444.81261109	96432668.2618988
6.6484375	4450794.062356	98298596.7351078
6.65234375	4491007.18628943	99245005.5384706
6.654296875	4511249.79154745	99721608.5527467
6.6552734375	4521405.28381459	99960764.4386777
6.65576171875	4526491.59980399	100080556.569678
6.655517578125	4523947.72698537	100020642.637482
6.6553955078125	**4522676.32674423**	99990699.0727337 – solution (4.52E6)

For the full sheet as a working demo, please see:

https://docs.google.com/spreadsheet/ccc?key=0Au6vC6P08mfGdHc0NW9KX0hEUmRqck1ldXdXY2VmRWc&hl=en_US#gid=0

Thanks,
Eddie Ma

Table 2: Solve for n in $n\lg n$ given $60E8$ total operations (1 minute).

$(10^x)\lg(10^x)$	n	$n\lg n$
4.88907562519182	77459.6669241484	1258034.66415039 – linear (lower bound)
9.77815125038364	6000000000	194893892128.245 – quadratic (upper bound)
7.33361343778773	21558246.7177851	525196326.28626
8.55588234408569	359651887.672942	10222035279.1448
7.94474789093671	88053757.0292739	2323904296.99322
8.2503151175112	177957017.091973	4877259706.77008
8.40309873079844	252987306.24633	7062011569.10496
8.32670692415482	212181211.189234	5869085426.14828
8.36490282747663	231687619.554538	6438044220.90344
8.34580487581573	221720003.009725	6147001657.1038
8.33625589998528	216898176.072285	6006440308.35024
8.33148141207005	214526729.578749	5937366659.87462
8.33386865602766	215709193.972772	5971803861.27806
8.33506227800647	216302868.068635	5989097107.37245
8.33565908899587	216600317.551225	5997762454.51174
8.33595749449058	**216749195.646809**	6002099816.96854 – solution (2.17E8)

Table 3: Solve for n in $n\lg n$ given $3600E8$ total operations (1 hour).

$(10^x)\lg(10^x)$	n	$n\lg n$
5.77815125038364	600000	11516761.7850948 – linear (lower bound)
11.5563025007673	360000000000	13820114142113.8 – quadratic (upper bound)
8.66722687557547	464758001.54489	13381267978.773
10.1117646881714	12934948030.671	434492085917.806
9.38949578187342	2451860639.76354	76476548737.3518
9.7506302350224	5631579703.2262	182411892888.274
9.93119746159689	8534880830.5572	281571896689.242
10.0214810748841	10507056675.9311	349786636372.653
10.0666228815278	11657968607.7157	389849434371.847
10.0440519782059	11067562373.3274	369276063352.241
10.032766526545	10783668444.5023	359399492946.634
10.0384092523755	10924693273.6261	364304367062.962
10.0355878894603	10853951820.4509	361843633561.647
10.0341772080027	10818753058.6636	360619496142.924
10.0334718672738	10801196506.2938	360008978639.342
10.0331191969094	**10792428916.9671**	359704106925.514 – solution (1.08E10)