Solutions to Practice Homework # 2

1. (a) True. If $f(n) = cg(n) + o(g(n))$ then

$$\lim_{n\to\infty} f(n)/g(n) = \lim_{n\to\infty} (cg(n) + o(g(n)))/g(n) = c.$$

Hence, by Claim 1 of Lecture 3, $f(n) = O(g(n))$.

(b) False. For example, if $f(n) = n^2$ and $g(n) = n$ then $f(n) \neq O(g(n))$. However,

$$\log f(n) = 2\log n = 2g(n) = O(g(n)).$$

(c) True, because $2^{\log_2 n} = n$, and $n = O(n^2)$.

2. (a) Working backwards, we show that $\sum_{k=0}^{n} x^k = (x^{n+1} - 1)/(x-1)$. First, multiply both sides by $x - 1$:

$$(x-1)(1 + x + x^2 + \ldots + x^n) = x^{n+1} - 1.$$

Thus,

$$x + x^2 + \ldots + x^n + x^{n+1} - (1 + x + x^2 + \ldots + x^n) = x^{n+1} - 1.$$

Most of the terms on the left hand side cancel out, leaving exactly those terms on the right hand side, as needed.

(b) Multiplying both sides the equation of part (a) by x^3 we get

$$\sum_{k=1}^{n} x^{k+2} = \frac{x^{n+3} - x^3}{x - 1}.$$

If we differentiate both sides of this equation we get:

$$\sum_{k=1}^{n} (k+2)x^{k+1} = \frac{(n+2)x^{n+3} - (n+3)x^{n+2} - 2x^3 + 3x^2}{(x-1)^2}.$$

When $x = 1/4$ and $n = 3$ we get .27, and when $x = 1/4$ and $n = 6$ we get .278 on both sides of the equation.

3. (a) $f(n) = 3n + 5$ and $g(n) = 3n$. Note that $5 = o(g(n))$ so $f(n) = g(n) + o(g(n))$.

(b) $f(n) = n$ and $g(n) = n^2$.

(c) $f(n) = 2n$ and $g(n) = n$. Note that although for this example, $f(n) = O(g(n))$, we have that $2^{f(n)} = 2^{2n} = 4^n$ but $4^n \neq O(2^n)$.

4. Let $R(n)$ be the number of runs of 0's in the set of binary strings of size n.

$$\left.\begin{matrix} 0 \\ 1 \end{matrix}\right\} \Rightarrow R(1) = 1$$

$$\left.\begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix}\right\} \Rightarrow R(2) = 3$$

If we append an n^{th} bit to the left all the strings of size $n - 1$ to get the set of all strings of size n we observe the following:

- For the 2^{n-1} strings where the added bit was 1 we have $R(n-1)$ runs of zeros
- For the 2^{n-1} strings where the added bit was 0 in half of the cases we appended the 0 to the left of a 1 which adds a total of 2^{n-2} runs of zeros to the preexisting $R(n-1)$ runs.

This means that $R(n) = 2R(n-1) + 2^{n-2}, n \geq 2$ and $R(1) = 1$.

We can solve for $R(n)$ as follows:

$$\begin{aligned} R(n) &= 2R(n-1) + 2^{n-2} \\ &= 4R(n-2) + 2 \times 2^{n-3} + 2^{n-2} = 4R(n-2) = 2^{n-2} + 2^{n-2} \\ &= 8R(n-3) + 2^{n-2} + 2^{n-2} + 2^{n-2} \\ &= \ldots \\ &= 2^i R(n-i) + i2^{n-2}. \end{aligned}$$

Setting $i = n-1$ we have that

$$R(n) = 2^{n-1} + n2^{n-2} = (n+1)2^n.$$

5. (a) The algorithm recursively call itself three times on three pairs of inputs, $(x_L, y_L), (x_H, y_H)$, and $(x_L + x_H, y_L + y_H)$. The size of inputs in first two pairs is $n/2$ since all of them are either upper half or lower half of n-bit number. The size of two inputs in the last pair is $\leq n/2 + 1$ since each of the two inputs is the sum of two $n/2$-bit numbers and the size of each sum is at most $n/2 + 1$ bits. Given that, we can formulate a recurrence relation of the running time of the algorithm:

$$M(n) = 2M(n/2) + M(n/2 + 1) + O(n).$$

The given inequality, namely $M(n) \leq 3M(n/2 + 1) + O(n)$, follows immediately from this.

(b) Let c be a constant such that $M(n) \leq 3M(n/2) + cn$. Then

$$\begin{aligned} M(n) &\leq 3M(n/2) + cn \\ &\leq 3(3M(n/2^2) + cn/2) + cn \\ &= 3^2 M(n/2^2) + (3/2)cn + cn \text{ (rearranging the terms of the previous line)} \\ &\leq 3^2(3M(n/2^3) + cn/2^2) + cn/2 + cn \\ &= 3^3 M(n/2^3) + (3/2)^2 cn + (3/2)cn + cn \\ &\leq \quad \ldots \\ &\leq 3^i M(n/2^i) + (3/2)^(i-1)cn + (3/2)^(i-2)cn + \ldots + (3/2)cn + cn. \end{aligned}$$

Let $i = \log_2 n$. Then the first term on the right hand side of the last line becomes $3^{\log_2 n} M(1)$ which is 0, since $M(1) = 1$. Therefore,

$$M(n) \leq cn \sum_{j=0}^{\log_2 n - 1} (3/2)^j.$$

We can apply the closed form expression of a geometric series here (see practice homework 2), where we let $x = 3/2$, to obtain

$$M(n) = cn \frac{(3/2)^{\log_2 n} - 1}{3/2 - 1}.$$

Since $(3/2)^{\log_2 n} = n^{\log_2 (3/2)}$, we get:

$$M(n) = 2cnn^{\log_2 (3/2)} - 2cn \leq 2cn^{1.6} = O(n^{1.6}).$$

Therefore, $M(n) = o(n^2)$.