Name: *Caleb McWhorter — Solutions*
MATH 308
Fall 2021
HW 19: Due 12/15

"Controlling complexity is the essence of computer programming."
–Brian Kernighan

Problem 1. (10pt) Prove that if $g(x)$ is $O(f(x))$, then $f(x)$ is $\Omega(g(x))$.

Solution. Suppose that $g(x)$ is $O(f(x))$. Then there exists $M, b \in \mathbb{R}$ such that $|g(x)| \leq M|f(x)|$ for $x \geq b$. But then for $x \geq b$, we know that $\frac{1}{M}|g(x)| \leq |f(x)|$. Therefore, choosing $M' = \frac{1}{M}$, there exist $M', b \in \mathbb{R}$ such that $|f(x)| \geq M'|g(x)|$ for $x \geq b$. But then $f(x)$ is $\Omega(g(x))$.

Problem 2. (10pt) Prove that if $f(x)$ is $O(g(x))$ and $c \in \mathbb{R} \setminus \{0\}$, then $cf(x)$ is $O(g(x))$.

Solution. Suppose that $f(x)$ is $O(g(x))$ and let $c \in \mathbb{R} \setminus \{0\}$. Because $f(x)$ is $O(g(x))$, there exist $M, b \in \mathbb{R}$ such that $|f(x)| \leq M|g(x)|$ for $x \geq b$. Observe that for $x \geq b$,

$$|cf(x)| = |c|\,|f(x)| \leq |c|M|g(x)|.$$

Choose $M' = |c|M$. Then there exists $M', b \in \mathbb{R}$ such that $|cf(x)| \leq M'|g(x)|$ for $x \geq b$. Therefore, $cf(x)$ is $O(g(x))$.

Problem 3. (10pt) Finding appropriate constants, show that $f(x) = 3x^4 + x^3 - 2x^2 + 6$ is $O(x^4)$.

Solution. By the triangle inequality, we know that

$$|f(x)| = |3x^4 + x^3 - 2x^2 + 6| \leq |3x^4| + |x^3| + |-2x^2| + |6| = 3|x^4| + |x^3| + 2|x^2| + 6.$$

If $x \geq 0$, then $x^4 \geq 0$. Similarly, for $x \geq 0$, we know $x^3 \geq 0$ and $x^2 \geq 0$. Now for $x \geq 1$, we know that $1 \leq 1 \cdot x = x$. But also $x \leq x \cdot x = x^2$, $x^2 \leq x^2 \cdot x = x^3$, and $x^3 \leq x^3 \cdot x = x^4$. Similarly, if $x \geq \sqrt[4]{6} > 1$, then we know $x^4 \geq 6$. But then for $x \geq \sqrt[4]{6}$, we know that

$$\begin{aligned}
|f(x)| = |3x^4 + x^3 - 2x^2 + 6| &\leq |3x^4| + |x^3| + |-2x^2| + |6| \\
&= 3|x^4| + |x^3| + 2|x^2| + 6 \\
&\leq 3x^4 + x^4 + 2x^2 + x^4 \\
&= 7x^4
\end{aligned}$$

Choosing $M = 7$ and $b = \sqrt[4]{6}$, there exists $M, b \in \mathbb{R}$ such that $|f(x)| \leq M|x^4|$ for $x \geq b$. Therefore, $f(x)$ is $O(x^4)$.

Problem 4. (10pt) Let $n \in \mathbb{Z}_{\geq 0}$. Find the number of operations (additions, subtractions, multiplications, and divisions) the following algorithm requires. What is the time complexity of the algorithm?

```
for i = 1 to n
  for j = 1 to i
  print(2n - i^2 j);
```

Solution. We assume that there are no flops required for printing. Given n, i, j, computing $2n - i^2 j$ requires 3 multiplications and 1 addition. Therefore, computing $2n - i^2 j$ requires 4 flops. This computation is required for each j from 1 to i. But then the integer $2n - i^2 j$ is computed a total of i times, each requiring 4 flops, which requires a total of $4i$ flops. This loops is computed for each i from 1 to n. We add the total number of flops required for each step of the iteration:

$$\begin{aligned} 4 \cdot 1 + 4 \cdot 2 + \cdots + 4 \cdot n &= \sum_{i=1}^{n} 4i \\ &= 4 \sum_{i=1}^{n} i \\ &= 4 \cdot \frac{n(n+1)}{2} \\ &= 2n(n+1) \\ &= 2n^2 + 2n. \end{aligned}$$

Clearly, this algorithm is $O(n^2)$. To see this definitively, observe that if $n \geq 1$, then $n \leq n \cdot n = n^2$. But then using this and the triangle inequality, for $x \geq 1$, $|2n^2 + 2n| \leq |2n^2| + |2n| = 2|n^2| + 2|n| \leq 2|n^2| + 2|n^2| = 4|n^2|$. Choosing $M = 4$ and $b = 1$, there exist $M, b \in \mathbb{R}$ such that $|2n^2 + 2n| \leq M|n^2|$ for $n \geq b$. Therefore, $2n^2 + 2n$ is $O(n^2)$.

Problem 5. (10pt) Let $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, where $a_i \in \mathbb{R}$, be a nonconstant polynomial.

(a) Compute the number of operations (additions, subtractions, multiplications, and divisions) required to compute $f(x_0)$ for some $x_0 \in \mathbb{R}$ the 'traditional way.'

(b) Horner's Method says to write $f(x)$ as...

$$a_0 + x\Big(a_1 + x\big(a_2 + x\big(a_3 + \cdots + x(a_{n-1} + a_n x)\big)\big)\Big)$$

Writing $f(x)$ as above, compute the number of operations required to compute $f(x_0)$.

Solution.

(a) Let $i \in \mathbb{Z}_{\geq 1}$. If one knows the values of a_0, $a_1 x$, $a_2 x^2$, ..., $a_n x^n$, then computing $f(x)$ simply requires n additions. Computing x^i requires $i - 1$ multiplications. Therefore, computing $a_i x^i$ requires i multiplications. But then computing $a_0, a_1 x, a_2 x^2, \ldots, a_n x^n$ requires

$$0 + 1 + 2 + \cdots + n = \sum_{i=0}^{n} i = \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

total multiplications. Therefore, computing $f(x)$ the 'traditional way' requires n additions and $\frac{n(n+1)}{2}$ multiplications for a total of...

$$n + \frac{n(n+1)}{2} = \frac{2n}{2} + \frac{n^2+n}{2} = \frac{n^2+3n}{2} = \frac{n(n+3)}{2}$$

operations, i.e. 'flops.' Therefore, this algorithm requires $O(n^2)$ total operations.

(b) Let $0 \leq i \leq n$. Computing $a_{i-1} + a_i x$ requires 1 multiplication and one addition. This computation is performed iteratively in Horner's method a total of n times for a total of n multiplications and n additions, i.e. a total of $2n$ operations or 'flops.' Therefore, evaluating $f(x)$ using Horner's method requires $O(n)$ total operations.

Clearly, evaluating $f(x)$ using Horner's method is much more computationally efficient than using the 'traditional way.' However, the 'traditional' way can be made more efficient as follows: to compute $x, x^2, x^3, \ldots, x^n$, begin with x and compute x^2, requiring one multiplication. To compute x^3, multiply x^2 by x, requiring one additional multiplication. Therefore, computing $x, x^2, \ldots, x^n$ requires a total of $0 + 1 + 1 + \cdots + 1 = n - 1$ multiplications. Computing $a_1 x, a_2 x^2, \ldots, a_n x^n$ then requires an additional n multiplications. Finally, to compute $f(x)$, we perform $n - 1$ additions. Therefore, computing $f(x)$ in the 'traditional way' using this approach requires $(n-1) + n = 2n - 1$ total multiplications and n additions for a total of $(2n-1) + n = 3n - 1$ total operations, i.e. flops. While this is still more total flops than Horner's method, computing $f(x)$ the 'traditional way' using this algorithmic approach requires $O(n)$ computations—identical to Horner's method.