We apply the definition of Big-O to MARIE programs. Please review the definition of Big-O below.

Big-O Notation

DEFINITION 1

Let T and f be functions from the set of integers or the set of real numbers to the set of real numbers. We say that $T(n)$ is $O(f(n))$ if there are positive constants n_0 and C such that

$$|T(n)| \leq C|f(n)|$$

whenever

$$n > n_0$$

[This is read as "$T(n)$ is big-oh of $f(n)$."]

- The constants C and n_0 in the definition of big-O notation are called **witnesses** to the relationship $T(n)$ is $O\big(f(n)\big)$.
- There are *infinitely many* witnesses to the relationship $T(n)$ is $O\big(f(n)\big)$.

Finding C, n_0, and $f(n)$ for $T(n)$

Steps: Assume $\boldsymbol{T(n) = \frac{3}{2}n^2 + \frac{5}{2}n + 10}$

1. Find $f(n)$. Let $f(n)$ be the fastest growing term in $T(n)$ with its coefficient removed.
   $\boldsymbol{f(n) = n^2}$
2. Find C.
   2.1. $C = C_{min} + \Delta$, where $\Delta = 1$ (in many cases).
   2.2. $C_{min} = \lim_{n\to\infty} \frac{T(n)}{f(n)} = \lim_{n\to\infty} \frac{\frac{3}{2}n^2 + \frac{5}{2}n + 10}{n^2} = \frac{3}{2}$
   2.3. In practice, C_{min} is the coefficient of the fastest growing term in $T(n)$.
3. $C = \Delta + \frac{3}{2} = 1 + \frac{3}{2} = \frac{5}{2}$
4. Find n_0.
   4.1. Solve $|T(n_0)| \leq C|f(n_0)|$

$$\frac{3}{2}n_0^2 + \frac{5}{2}n_0 + 10 \leq \frac{5}{2}n_0^2$$

$$n_0^2 - \frac{5}{2}n_0 - 10 \geq 0$$

$$n_0 = \left\lceil \frac{\frac{5}{2} \mp \sqrt{\left(\frac{5}{2}\right)^2 + 4 \cdot 1 \cdot 10}}{2} \right\rceil, n_0 > 0$$

$$n_0 = \lceil 4.65 \rceil, n_0 > 0$$

$$n_0 = 5$$

   4.2. Choose an integer value for n_0. Let $n_0 = 5$.

We have shown that $\boldsymbol{T(n) = \frac{3}{2}n^2 + \frac{5}{2}n + 10}$ is $\boldsymbol{O(n^2)}$ because we have found witnesses $\boldsymbol{C = \frac{5}{2}}$ and $n_0 = 5$.

Now we apply the definition to a MARIE program that divides a positive integer by two.

```
            input
            store dividend
test,       subt  divisor
            skipcond 000
            jump inc
            load quotient
            output
            halt
inc,        store dividend
            load quotient
            add one
            store quotient
            load dividend
            jump test
dividend,   dec 0
divisor,    dec 2
quotient,   dec 0
one,        dec 1
```

Each instructions executed is assigned unit (1) cost. Instructions in black are executed only once. Instructions in blue are executed, possibly, more than once. The question is, how many times are the instructions shown in blue executed? If we examine the code, we find that the loop adds one to the quotient every time two is subtracted from the dividend – in effect, this code divides a positive integer value by two. Thus, the code in blue is executed half as many times as the input value stored in the dividend. We will call the dividend n to be consistent with our terminology for time complexity.

Line	Code		Cost
1		input	1
2		store dividend	1
3	test,	subt divisor	$n/2$
4		skipcond 000	$n/2$
5		jump inc	$n/2$
6		load quotient	1
7		output	1
8		halt	1
9	inc,	store dividend	$n/2$
10		load quotient	$n/2$
11		add one	$n/2$
12		store quotient	$n/2$
13		load dividend	$n/2$
14		jump test	$n/2$

Line	Code	Cost
15	**dividend, dec 0**	0
16	**divisor, dec 2**	0
17	**quotient, dec 0**	0
18	**one, dec 1**	0
	Total	$T(n) = \frac{9}{2}n + 5$

1. **Find $f(n) = n$**
2. **Find $C = C_{min+}\Delta = \frac{9}{2} + 1 = \frac{11}{2}$**
3. **Solve for n_0 $|T(n_0)| = C|f(n_0)|$**
   $\left|\frac{9}{2}n_0 + 5\right| = \frac{11}{2}|n_0|$
   $5 = n_0$

We have shown that $T(n) = \frac{9}{2}n + 5$ is $O(n)$ because we have found witnesses $C = \frac{11}{2}$ and $n_0 = 5$.