Exercise 1: Asymptotic Analysis, Code Modeling

Due date: Friday July 12, 2019 at 11:59 pm

Instructions:

Submit a typed or neatly handwritten scan of your responses to the "Exercise 1" assignment on Gradescope here: https://www.gradescope.com/courses/52323. Make sure to log in to your Gradescope account using your UW email to access our course.

For more details on how to submit, see
https://courses.cs.washington.edu/courses/cse401/18au/hw/submitting_hw_guide.pdf.

These problems are meant to be done **individually**. If you do want to discuss problems with a partner or group, make sure that you're writing your answers individually later on. Check our course's collaboration policy if you have questions.

1. Asymptotic analysis: Mathematically

(a) Show that $10\log_2(n) + 15 \in \mathcal{O}(n\log_2(n))$ is true by finding a c and n_0 that satisfy the definition of "dominated by" and big-$\mathcal{O}$. Please show your work.

(b) Show that $\log_2(n) \in \mathcal{O}(\log_8(n))$ by finding a c and n_0 that satisfy the definition of "dominated by" and big-$\mathcal{O}$. Please show your work. As a hint, you will need to use the change-of-base logarithm identity somewhere.

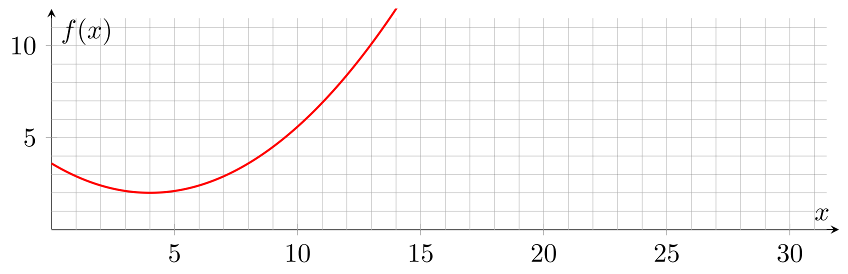
(c) Let $a \geq 2$ and $b \geq 2$ be constants. Is it always true that $\log_b(n) \in \mathcal{O}(\log_a(n))$? Explain why. (Hint: was anything special about 2 and 8 in your last proof?)

2. Asymptotic analysis: Visually

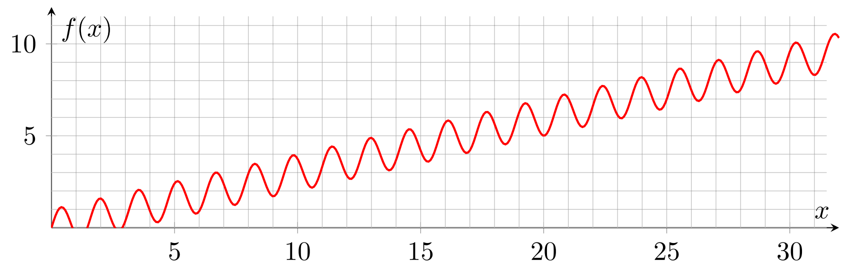
For each of the following plots, provide a tight big-$\mathcal{O}$ bound, a tight big-Ω bound, and a big-Θ bound. You do not need to show your work; just list the bounds. If a particular bound doesn't exist for a given plot, briefly explain why. Assume that the plotted functions continue to follow the same trend shown in the plots as x increases. Each provided bound must either be a constant or a simple polynomial, **from the following possible answers**.

$$n^2, 1, n, log(n), n!, 1/n$$

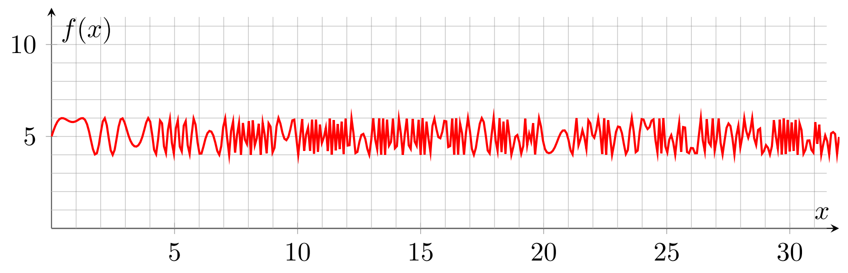
(a)



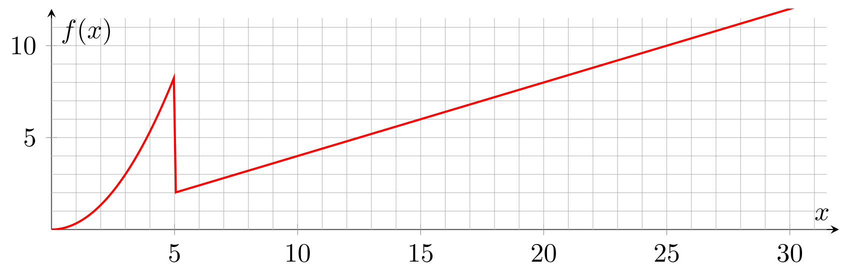
(b)



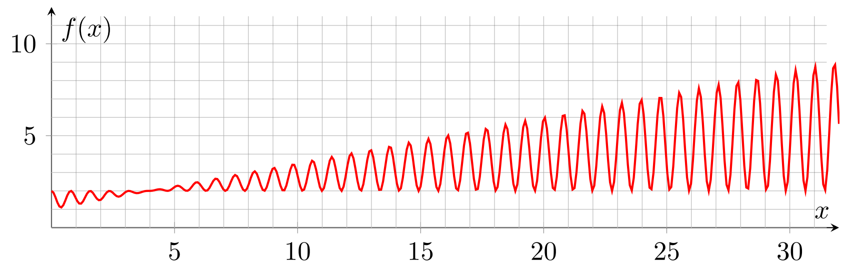
(c)



(d)



(e)



3. Modeling code

In this problem we will analyze code in the countProduct method below. The code is not particularly efficient (i.e. you should not use this code snippet as a model for how to use data structures).

The DoubleLinkedList object below is the one you implement in Project 1 – be sure to understand the relevant running times from there.

```
 1     public void countProduct(DoubleLinkedList<Integer> input1, DoubleLinkedList<Integer> input2,
 2                              DoubleLinkedList<Integer> toFind) {
 3
 4     DoubleLinkedList<Integer> products = new DoubleLinkedList<>();
 5
 6     for (int n1 : input1) {
 7         for (int n2 : input2) {
 8             int currProduct = n1 * n2 + 373;
 9             products.add(currProduct);
10         }
11     }
12
13     DoubleLinkedList<Integer> results = new DoubleLinkedList<>();
14
15     for (int target : toFind) {
16         if (products.contains(target)) {
17             int location = products.indexOf(target);
18
19             if (target % 2 == 0) {
20                 results.add(location);
21             } else {
22                 helperFunction(toFind);
23             }
24         } else {
25             if (target % 2 == 0) {
26                 System.out.println("target not found");
27             } else {
28                 helperFunction(toFind);
29             }
30         }
31     }
32 }
```

Answer the following questions about the runtime of the countProduct method. In this problem, assume input1 has n elements, input2 has m elements, and toFind has p elements. Note: the code above calls the helperFunction method. You should assume that on an input of size k, helperFunction always runs in $\Theta(k)$ time. Also assume that helperFunction does not alter its input.)

Remember that "best/worst-case" refer to the inputs that yield the fastest or slowest possible runtime, respectively.

In this problem, you will be working with code modeling concerning multiple variables. We recommend that you check out this helpful document before you proceed: https://courses.cs.washington.edu/courses/cse373/19su/files/resources/multivariableBigO.pdf.

(a) What is the simplified tight big-$\mathcal{O}$ bound for both the best-case and worst-case runtime for the loop in lines 6 - 11? Your answer should be in terms of n, m and/or p.

(b) What is the simplified tight big-$\mathcal{O}$ bound for worst-case runtime for lines 17 - 23? Your answer should be in terms of n, m, and/or p. (Hint: Think about how big `products` can be)

(c) What is the simplified tight big-Ω bound for worst-case runtime for line 17 - 23? Your answer should be in terms of n, m, and/or p.

(d) Answer the following about the best case now:

- Describe how `input1`, `input2`, and `toFind` could look to achieve your best case runtime for line 17 - 23. Advice: since we are doing asymptotic analysis, your best case scenario(s) should work for any arbitrary input size for all three lists.

- What is the simplified tight big-$\mathcal{O}$ bound for best-case runtime for line 17 - 23? Your answer should be in terms of n, m, and/or p.

(e) What is the simplified, tight big-$\mathcal{O}$ bound for the worst-case runtime for the loop in lines 15 - 31? Your answer should be in terms of n, m, and/or p.

(f) What is the simplified tight big-$\mathcal{O}$ bound for the worst-case runtime of `countProduct`? Your answer should be in terms of n, m, and/or p.