Homework 1: Analysis

COS 226

In case this is useful:

- Multiplication by a constant:
  - $f(n) \in \mathcal{O}(g(n)) \Rightarrow cf(n) \in \mathcal{O}(g(n)),\ c > 0$
- Addition:
  - $a(n) \in \mathcal{O}(f(n))\ \&\ b(n) \in \mathcal{O}(g(n)) \implies a(n) + b(n) \in \mathcal{O}(f(n) + g(n))$
- Multiplication:
  - $a(n) \in \mathcal{O}(f(n))\ \&\ b(n) \in \mathcal{O}(g(n)) \implies a(n)b(n) \in \mathcal{O}(f(n)g(n))$
- Transitivity:
  - $a(n) \in \mathcal{O}(f(n))\ \&\ f(n) \in \mathcal{O}(g(n)) \implies a(n) \in \mathcal{O}(g(n))$
- Polynomials:
  - $f(n)$ is a polynomial of degree $d \implies f(n) \in \mathcal{O}(n$
- Exponential bound on polynomial:
  - $n^x \in \mathcal{O}(a^n)$ for any fixed $x > 0,\ a > 1$
  - E.g., $n^{1000} \in \mathcal{O}(2^n)$
- Log of power:
  - $\log n^x \in \mathcal{O}(\log n)$ for any fixed $x > 0$
  - Because: $\log n^x = x \log n \in \mathcal{O}(\log n)$
- Power of log:
  - $\log^x n = (\log n)^x \in \mathcal{O}(n^y)$ for any fixed $x > 0,\ y > 0$

1. Let $f(n) = (n+3)(n^2+1)$
   (a) Find $g(n)$ such that $f(n)$ is $\mathcal{O}(g(n))$.
   (b) What are c and n_0 that shows your answer is correct?

2. If $f(n) = n^{1000} + 3n^2$ and $g(n) = 2^n$, is $f(n) \in o(g(n))$? Why or why not?

3. Suppose $f(n) = (\log^6 n)(\log n^3)$. Show that $f(n)$ is $\mathcal{O}(n \log n)$.

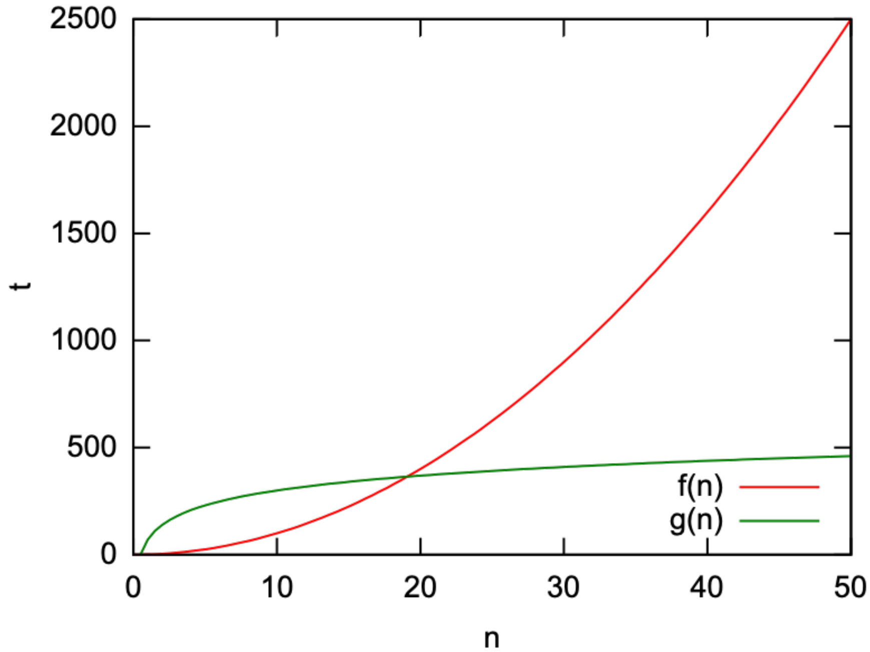
4. Suppose we have the following algorithm:

```
1: Algorithm Cartesian(A, B, n)
2:     Input: A and B, two n-element lists
3:     Output: The Cartesian product of the two lists: [[A[0], B[0]], [A[0], B[1]], ...]
4:     Let C be an empty list
5:     for i from 0 to n − 1 do
6:         for j from 0 to n − 1 do
7:             Add [A[i], B[j]] to the end of C
8:     return C
9: End.
```

   (a) What is the time complexity using the RAM model (i.e., directly counting operations)? Make sure you explain your answer in terms of the operations you consider primitive.
   (b) Is this algorithm's running time $\mathcal{O}(n^3)$? Why or why not?
   (c) Is this algorithm's running time $\Theta(n^2)$? Why or why not?
   (d) Is this algorithm's running time $\Omega(n)$? Why or why not?

5. Suppose that we add a loop between lines **??** and **??** of the algorithm in question **??** that prints the list, one element at a time. Which of the answers you gave in question **??** would be different now? Why?

6. Given the graph below:



what can you say about the relationship between $f(n)$ and $g(n)$? Make sure you reference c and n_0 in your answers.