Definitions

Consider two functions $f(n)$ and $g(n)$.

- $f(n) = O(g(n))$ if there exists constants $c > 0, n_0 \geq 0$ such that $\forall n \geq n_0 : f(n) \leq c \cdot g(n)$.
- $f(n) = \Omega(g(n))$ if there exists constants $d > 0, n_0 \geq 0$ such that $\forall n \geq n_0 : d \cdot g(n) \leq f(n)$.
- $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ <u>and</u> $f(n) = \Omega(g(n))$.

Questions

1. Let $f(n) = 3n^2 + 2n + 40$ and let $g(n) = n^2$. Prove that $f(n)$ is $O(g(n))$.

2. Let $f(n) = 3n^2 + 2n + 40$ and let $g(n) = n^2$. Prove that $f(n)$ is $\Omega(g(n))$.

3. For each of the following, determine if a particular type of function exists. If it does, give an example.

   (a) Is there a function $f(n)$ where $f(n) = O(n^2)$ but $f(n) \neq \Omega(n)$?

   (b) Is there a function $f(n)$ where $f(n) = \Omega(n^2)$ but $f(n) \neq O(n)$?

   (c) Is there a function $f(n)$ where $f(n) = O(n)$ but $f(n) \neq O(n^2)$?

   (d) Is there a function $f(n)$ where $f(n) \neq O(n)$ but $f(n) \neq \Omega(n)$?

4. *(Kleinberg-Tardos 2.3)* Arrange the following list of functions in **ascending** order of growth rate. That is, if $f(n)$ precedes $g(n)$ in your list, then $f(n) = O(g(n))$. Make sure to identify any ties – $f(n)$ and $g(n)$ where $f(n) = \Theta(g(n))$.

$$n^{2.5} \qquad \sqrt{2n} \qquad n + 10 \qquad 10^n \qquad 100^n \qquad n^2 \log n$$

5. Imagine n students live in a dorm. You want to identify a group of k students who live in the dorm, but have never taken a class together. But, you don't know for sure that such a group exists.

   (a) How many different groups of k students are there of the n students living in the dorm?

   (b) Let's assume that k is constant (can't change). Make an argument that to loop through all such groups would be $O(n^k)$.

   (c) A brute-force algorithm to answer our question is to loop through all groups of k students and for each group, consider all pairs of students to determine if they have taken a class together. Assume that it takes constant time $O(1)$ to determine if any two students have taken a class together. What would the asymptotic runtime of this algorithm be?