Big-Oh Notation

Exercise 1

- Prove that the Fibonacci algorithm (for-loop version) is $O(n)$.

Exercise 1

- Prove that the Fibonacci algorithm (for-loop version) is $O(n)$.

Recall that $f(n) = 6n$, and thus we find $c = 6$ such that

$f(n) \leq 6 \cdot n$ when $n > 0$.

Exercise 1

- Prove that the Fibonacci algorithm (for-loop version) is $O(n^2)$.

Exercise 1

- Prove that the Fibonacci algorithm (for-loop version) is $O(n^2)$.

$f(n) = 6n \leq c \cdot n^2$ (pick $c = 1$)

→ $6n \leq n^2$ (omit n from both sides)

→ $6 \leq n$ ($n_0 = 6$)

So that $f(n) \leq n^2$ when $n \geq 6$.

Exercise 1

- Which one should we pick for the Fibonacci algorithm? $O(n)$ or $O(n^2)$? Why?

Exercise 1

- Which one should we pick for the Fibonacci algorithm? $O(n)$ or $O(n^2)$? Why?

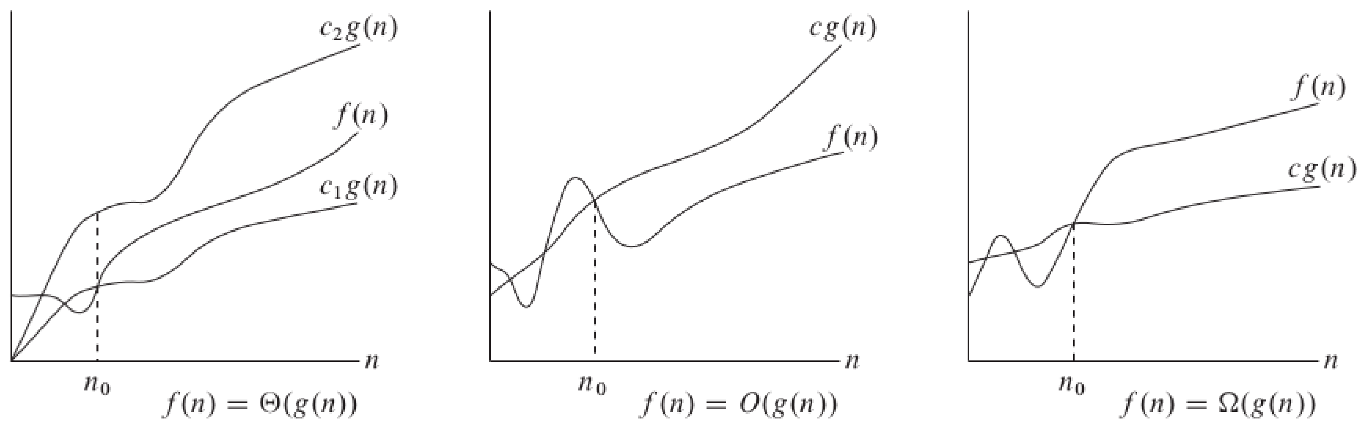
$O(n)$. Because Big-Oh defines the upper bound of an algorithm, and we want to limit the upper bound as small as possible.

Think about a set $S = \{5, 14, 23, 40\}$ and upper bounds 100000000 vs. 40

By using $O(n)$, we know that for the worst-case scenario, the Fibonacci algorithm takes linear time.

Exercise 2

- Suppose that an algorithm A for some problem takes time asymptotic to n^2 with one input I_0 of size n, where there are many other inputs of sizes n. Use *Big-X* (Θ, Ω, O) notations to answer the following questions:
  - *Q1. How would you describe A's best-case complexity?*
  - *Q2. How would you describe A's worst-case complexity?*
  - *Q3. How would you describe A's average-case complexity?*



Exercise 2

Q1. We know of one input that results in n^2 , but there may be better ones. Suppose $f(n)$ is the best-case complexity of A, then $f(n) \leq n^2$ for any n, n^2 provides an upper bound of $f(n)$.
Big-O notation applies: $f(n) \leq cn^2$ and $f(n) = O(n^2)$.

Q2. In opposition to Q1, suppose $f(n)$ is the worst-case complexity of A. we know $f(n) \geq n^2$ for any n. n^2 provides a lower bouncomplexityd of $f(n)$.
Big-Omega notation applies: $f(n) \geq cn^2$ and $f(n) = \Omega(n^2)$.

Q3. It cannot be determined as we only have the information for one input. We need more information (probability distribution, etc.) of the inputs to answer the average-case complexity.