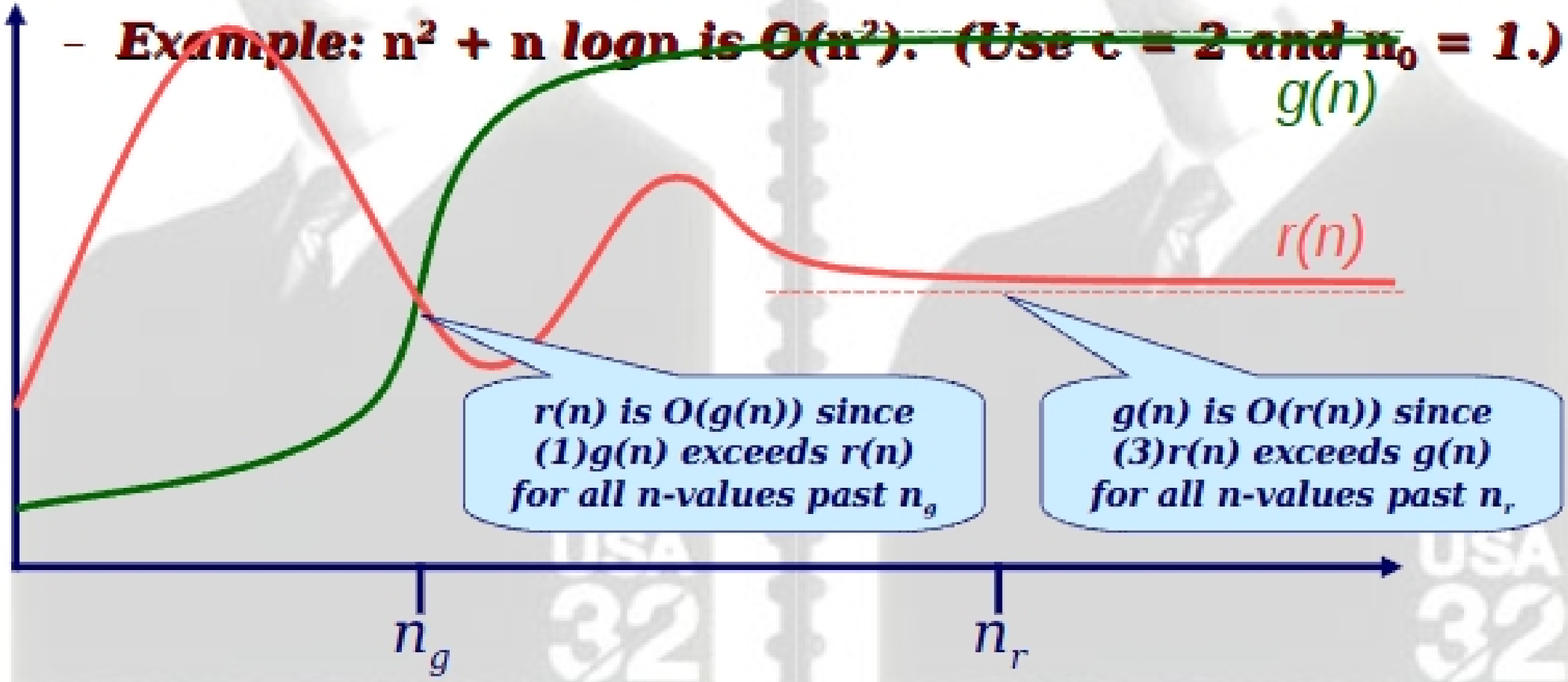
Chapter 9
Algorithm Efficiency & Sorting

Programming efficiency has been important up to this point, but we're now going to examine the methods that have been developed to determine just how efficient a program is.

- ✓ Measuring Algorithm Efficiency
- ✓ Example: Searching Algorithms
- ✓ Example: Sorting Algorithms

Time Complexity Terminology: Big-O

- **Function T(n) is said to be O(f(n)) if there are positive constants c and n_0 such that $T(n) \leq c\ f(n)$ for every $n \geq n_0$.**
  - **Example: $n^3 + 3n^2 + 6n + 5$ is $O(n^3)$. (Use $c = 15$ and $n_0 = 1$.)**
  - **Example: $n^2 + n$ loop is $O(n^2)$. (Use $c = 2$ and $n_0 = 1$.)**



Demonstrating The Big-O Concept

Both algorithms below have $O(n^3)$ time complexity.

		ALGORITHM	
		A	B
Input Size n	10	1,110	11,110
	100	1,010,100	2,010,100
	1,000	1,001,001,000	1,101,001,000
	10,000	1,000,100,010,000	1,010,100,010,000
	100,000	1,000,010,000,100,000	1,001,010,000,100,000
	1,000,000	1,000,001,000,001,000,000	1,000,101,000,001,000,000

(In fact, the execution time for Algorithm A is $n^3 + n^2 + n$, and the execution time for Algorithm B is $n^3 + 101n^2 + n$.)