Algorithm Efficiency

Assume that the function $g(n)$ is the time required to run Algorithm A.

An algorithm $A$ is said to be “of order $f(n)$,” denoted by $O(f(n))$, if constants $c_1$, $c_2$, and $n_0$ can be found such that:

$$c_1 \times t(n) < g(n) < c_2 \times t(n)$$

for a problem of size $n > n_0$

---

Order Analysis Example

- A: initialization code executed once
- while iteration; no. of iterations depends on $n$
- B: linear sequence of stmts
- while iteration, dependent on $n$
- C: linear sequence
- D: linear sequence
- E: linear sequence
- while iteration, not dependent on $n$
- F: linear sequence
Order Analysis Example

- $c_2 = 10$
  - $T(n) = 10n^2$

- $g(n) = 4n^2 + 2n^2 + 4$

- $c_1 = 1$
  - $T(n) = n^3$

Running Time vs. Size of problem $n$