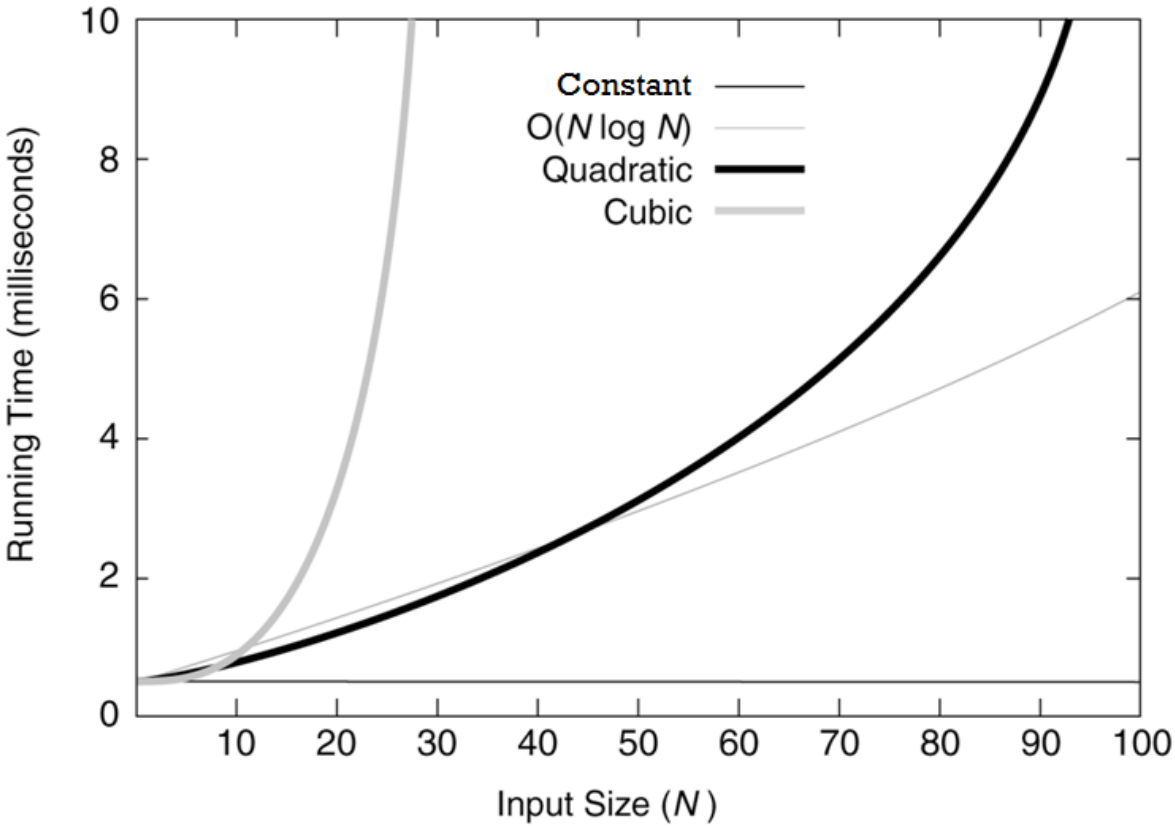
Example Ch 2

Complete:

Function	Growth Rate Name
c	**Constant**
$\log N$	**Logarithmic**
$\log^2 N$	**Log-squared**
N	**Linear**
$N \log N$	**N log N**
N^2	**Quadratic**
N^3	**Cubic**
2^N	**Exponential**



- $O(1)$ Time requirement is constant, and it is independent of the problem's size.
- $O(\log_2 n)$ Time requirement for a logarithmic algorithm increases slowly as the problem size increases.
- $O(n \log_2 n)$ Time requirement for a $N \log_2 N$ algorithm increases more rapidly than a linear algorithm.
- $O(n^3)$ Time requirement for a cubic algorithm increases more rapidly with the size of the problem than the time requirement for a quadratic algorithm.

If an algorithm takes 2 seconds to run with the problem size 4, what is the time requirement (approximately) for that algorithm with the problem size 8, if its order is:

O(1) ➔ T(n) =**2 seconds**

O(log2n) ➔ T(n) =**2*log(8)/log(4)) = 3 seconds**

O(n) ➔ T(n) = **(2*8) / 4= 4 seconds**

O(n log2n) ➔ T(n) =**2*8*log(8)/(4*log(4)) = 6 seconds**

O(n^2) ➔ T(n) = **(2*8^2) / 4^2= 8 seconds**

O(n^3) ➔ T(n) = **(2*8^3) / 4^3= 16 seconds**

O(2^n) ➔ T(n) = **(2*2^8) / 2^4= 32 seconds**