2017-09-21

- Asymptotic notations:
  - $f(n) = \mathrm{O}(g(n))$: upper bound. $\exists\, k > 0, n_0 > 0$ such that $f(n) \le k \cdot g(n)\ \forall\, n > n_0$
  - $f(n) = \Theta(g(n))$: tight bound. $\exists\, k_1 > 0, k_2 > 0, n_0 > 0$ such that $k_1 \cdot g(n) \le f(n) \le k_2 \cdot g(n)\ \forall\, n > n_0$
  - $f(n) = \Omega(g(n))$: lower bound. $\exists\, k > 0, n_0 > 0$ such that $k \cdot g(n) \le f(n)\ \forall\, n > n_0$
  - $f(n) = \mathrm{o}(g(n))$: strict upper bound. $\forall\, k > 0\ \exists\, n_0 > 0$ such that $f(n) < k \cdot g(n)\ \forall\, n > n_0$
  - $f(n) = \omega(g(n))$: strict lower bound. $\forall\, k > 0\ \exists\, n_0 > 0$ such that $k \cdot g(n) < f(n)\ \forall\, n > n_0$
- The property of $f(n) = \mathrm{O}(g(n))$ holds after summation, multiplication, and power, but not logarithm and exponential.
- Average-case analysis requires the assumption about the probability distribution of problem instances.
- Time complexity and space complexity focus on the worst-case complexity.
- The (worst-case) time complexity of an algorithm is said to be $\Theta(f(n))$ if $\exists\, f(n)$ s.t. A runs in time $O(f(n))$ and $\Omega(f(n))$.
- Algorithm A is *no worse than* Algorithm B in terms of worst-case time complexity if $\exists\, f(n)$ s.t. A runs in time $O(f(n))$ and B runs in time $\Omega(f(n))$.
- Algorithm A is *(strictly) better than* Algorithm B in terms of worst-case time complexity if $\exists\, f(n)$ s.t. either
  - A runs in time $O(f(n))$ and B runs in time $\omega(f(n))$.
  - A runs in time $o(f(n))$ and B runs in time $\Omega(f(n))$.
- The (worst-case) time complexity of a problem is said to be $\Theta(f(n))$ if *both*
  - The time complexity of the problem is $O(f(n))$, i.e. there exists an $O(f(n))$-time algorithm that solves the problem.
  - The time complexity of the problem is $\Omega(f(n))$, i.e. any algorithm that solves the problem requires $\Omega(f(n))$ time.
- Problem P is *no harder than* Problem Q in terms of (worst-case) time complexity if $\exists\, f(n)$ s.t. the time complexity of P is $O(f(n))$ and that of Q is $\Omega(f(n))$.
- Problem P is *(strictly) easier than* Problem Q in terms of (worst-case) time complexity if $\exists\, f(n)$ s.t. either
  - The time complexity of P is $O(f(n))$ and that of B runs in time $\omega(f(n))$.
  - The time complexity of P is $o(f(n))$ and that of B runs in time $\Omega(f(n))$.