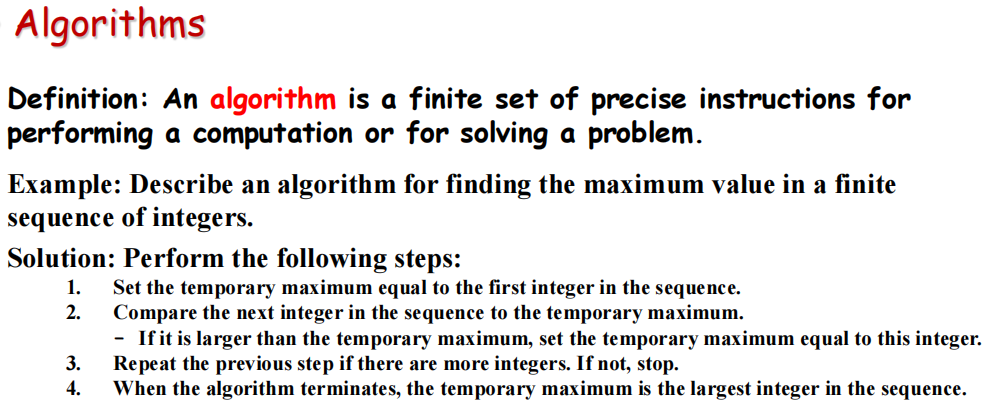
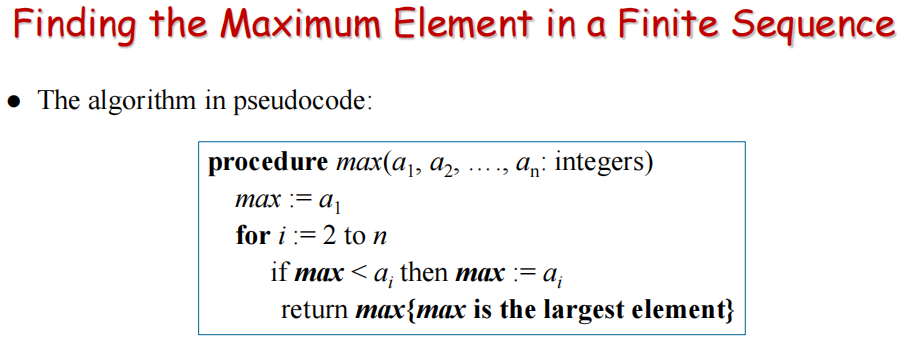
**（3.1）Algorithms**

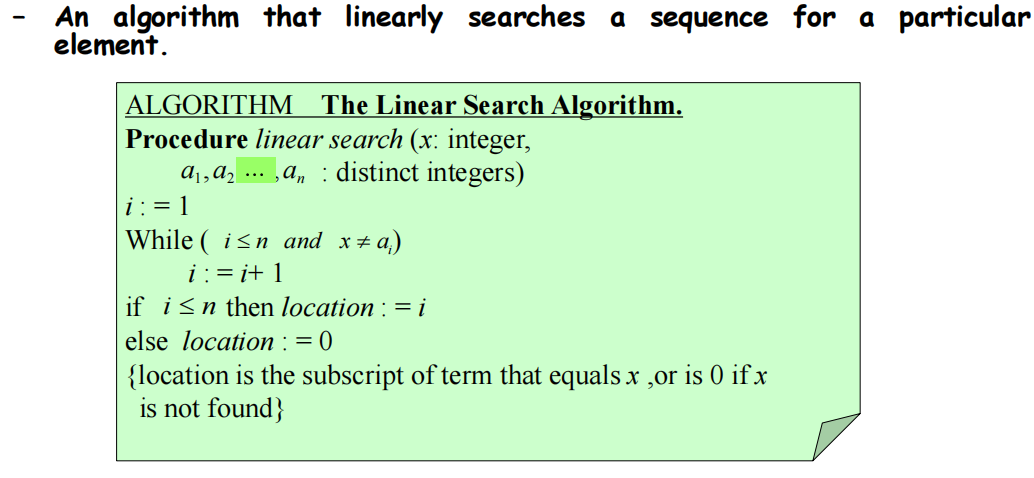


**Pseudocode（伪码）**



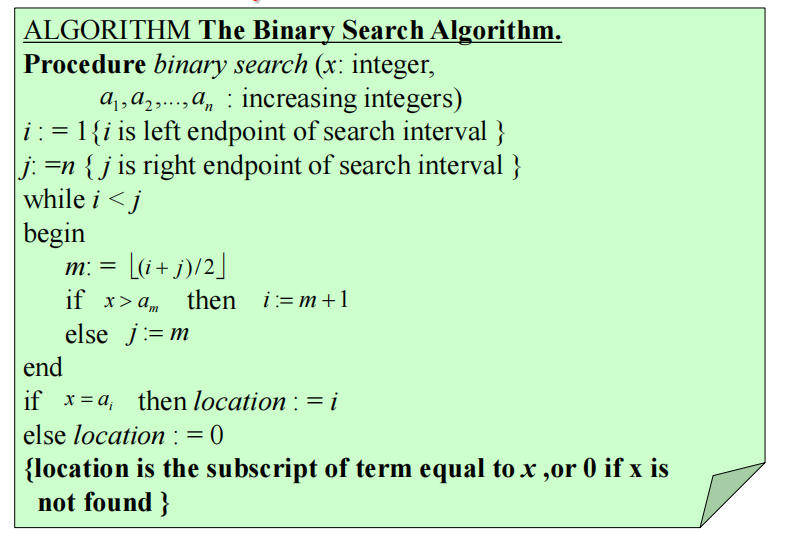
Searching Problems

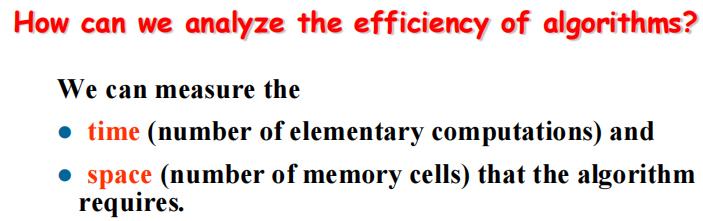
**Linear Search or sequential search（线性搜索或顺序搜索）**



**Binary Search（折半搜索）**

**（前提，对数据先排序）**

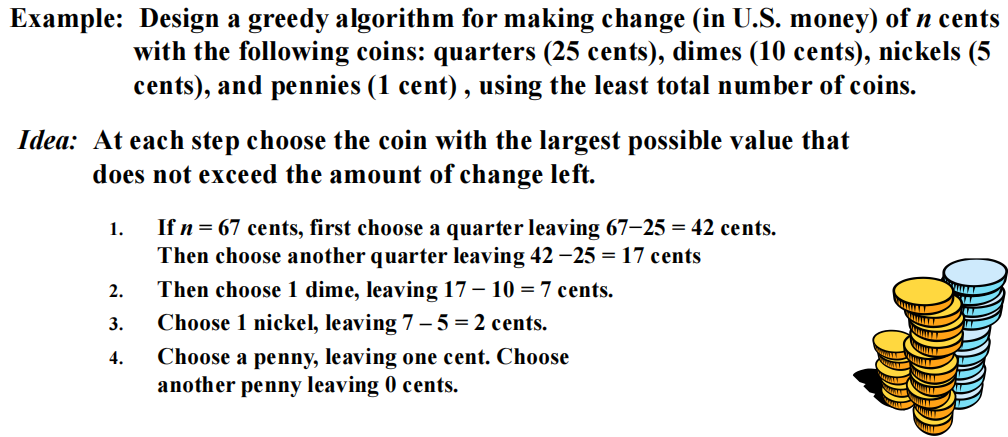


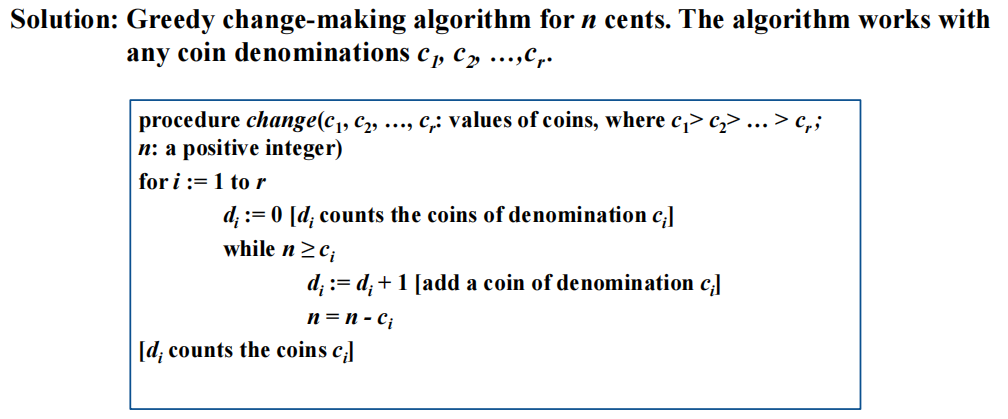


**These measures are called computational complexity（计算复杂度） and space complexity（空间复杂度）, respectively.**

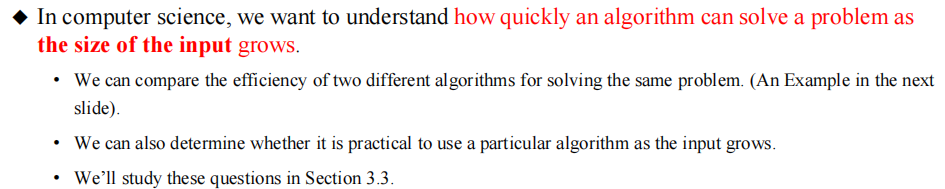
Greedy Algorithms（贪心算法）

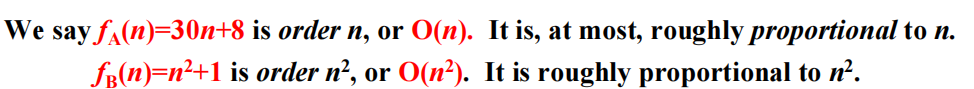
Example：Making Change

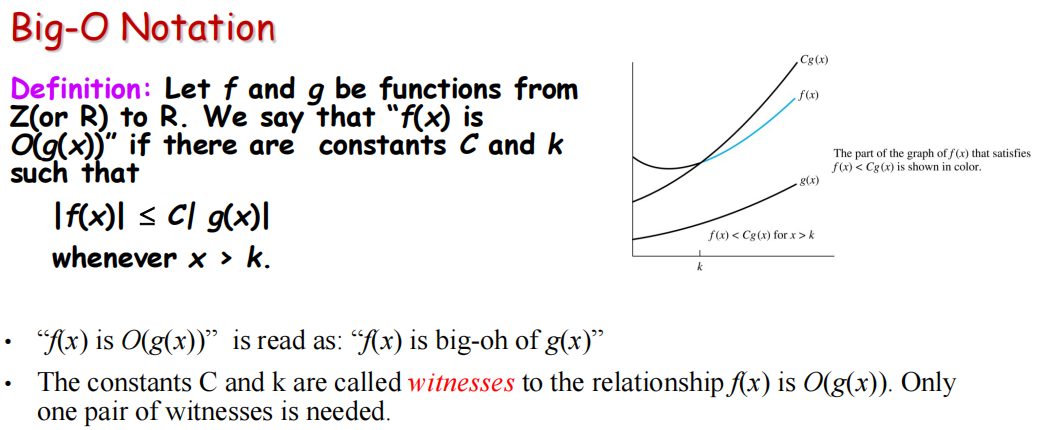


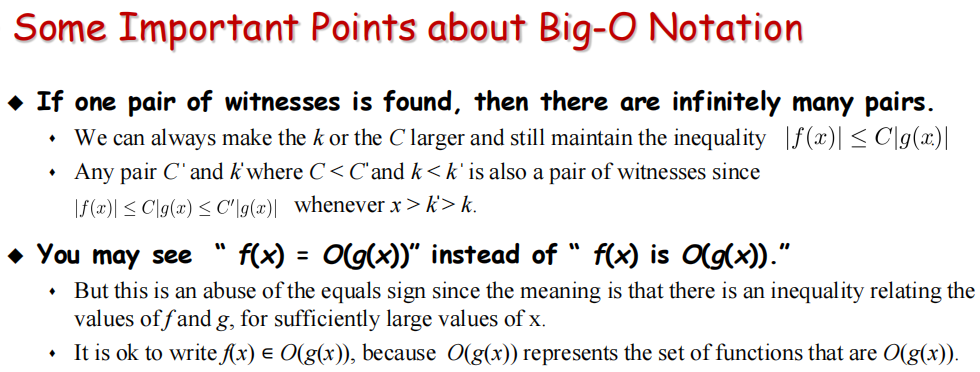


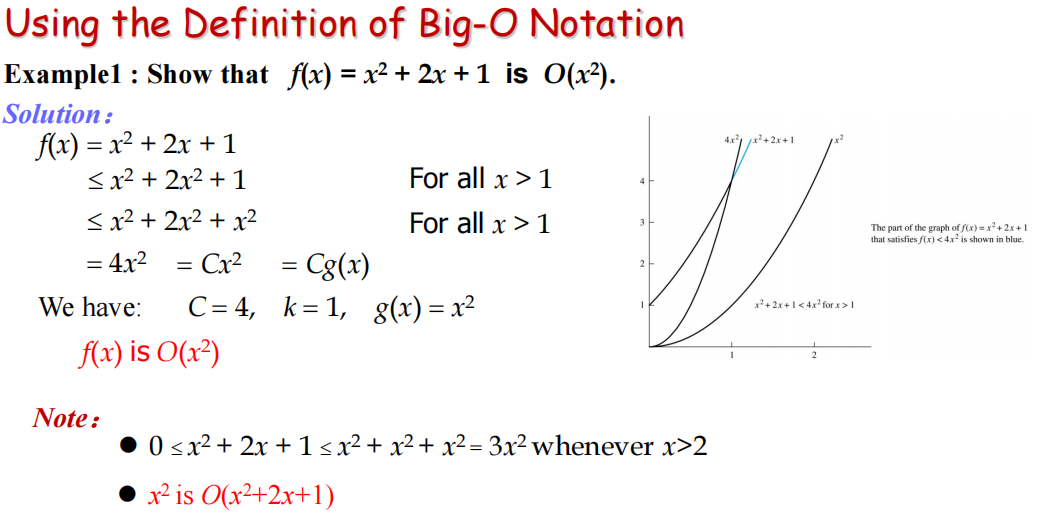
**（3.2）The Growth of Functions（函数的增长率）**

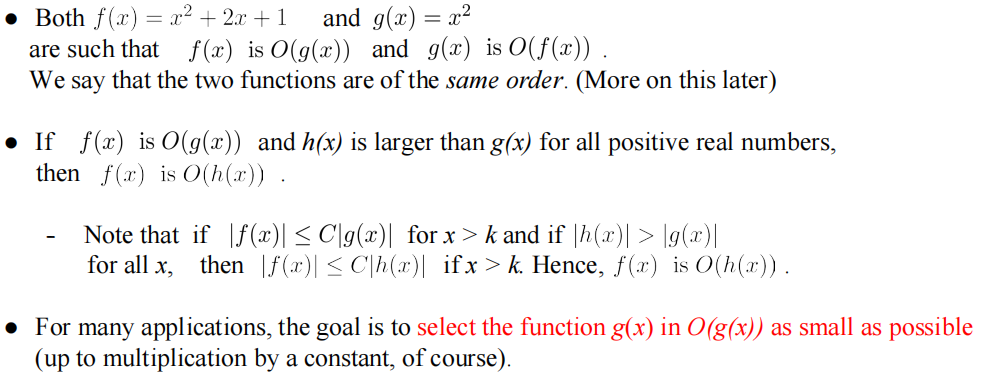


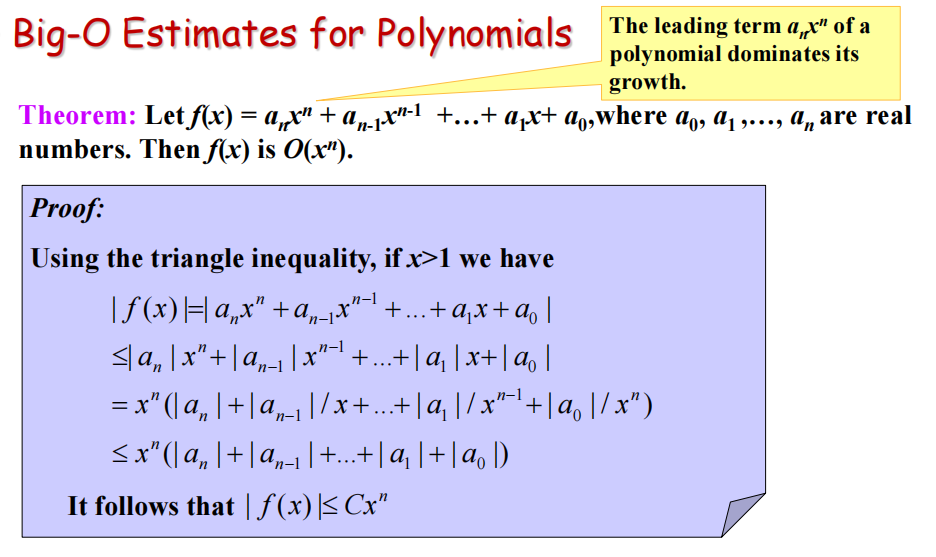


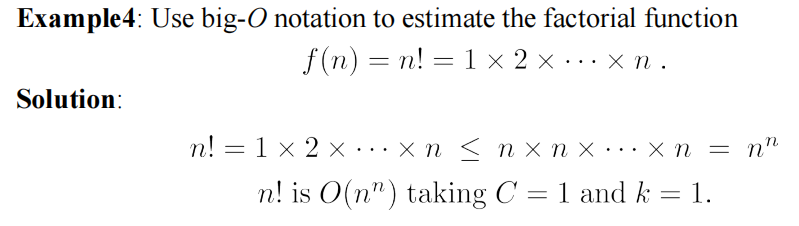


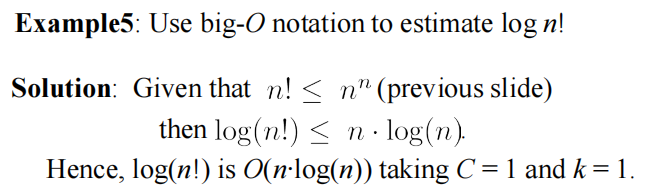


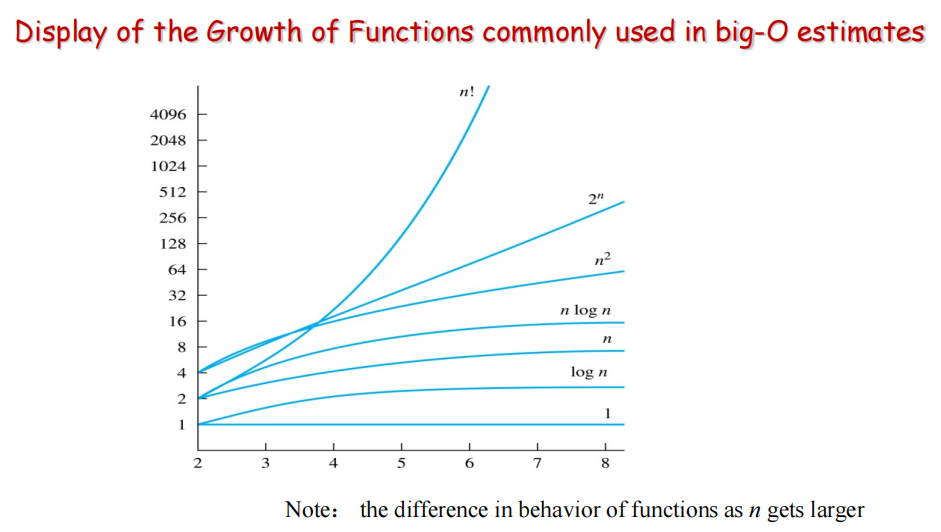


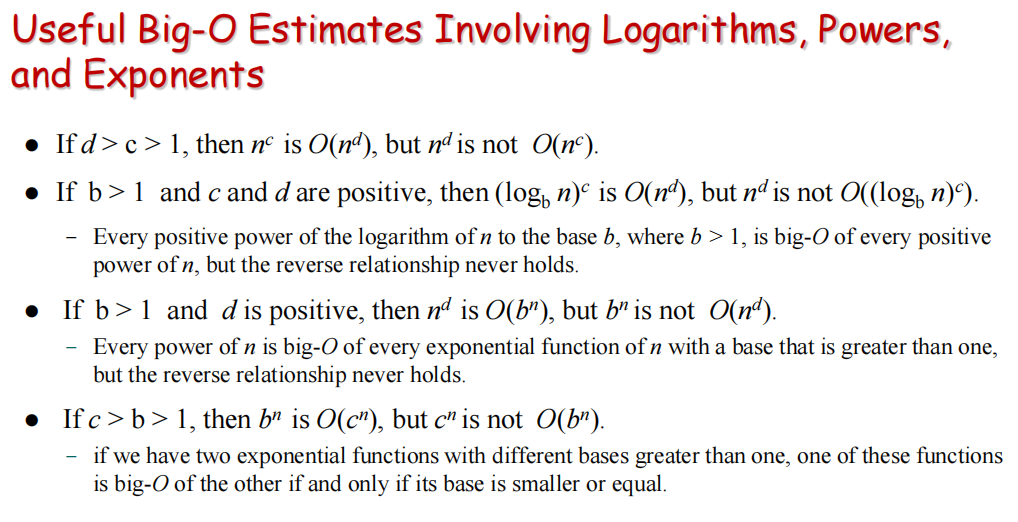


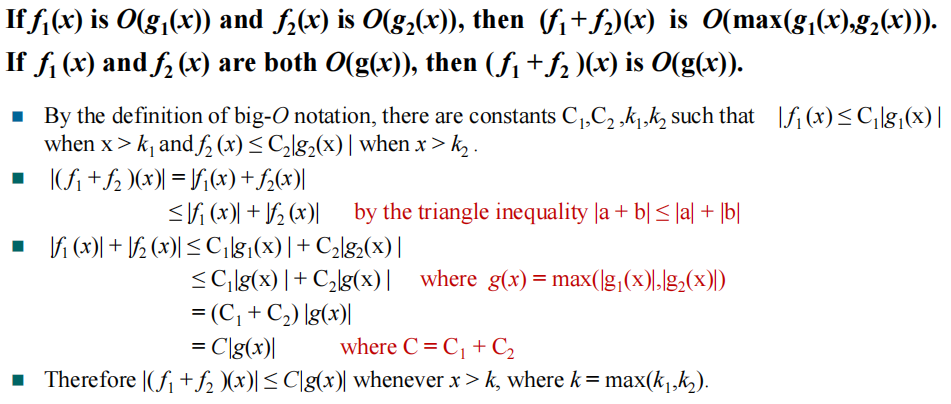


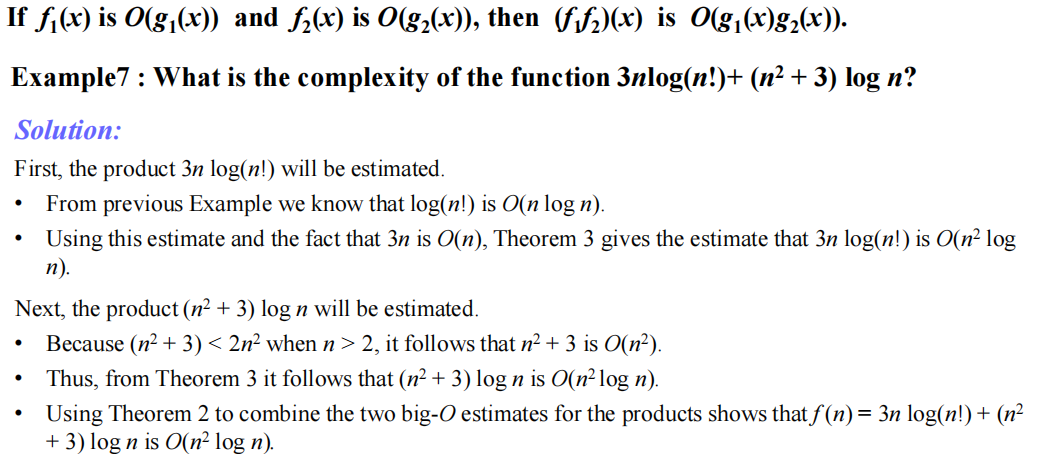


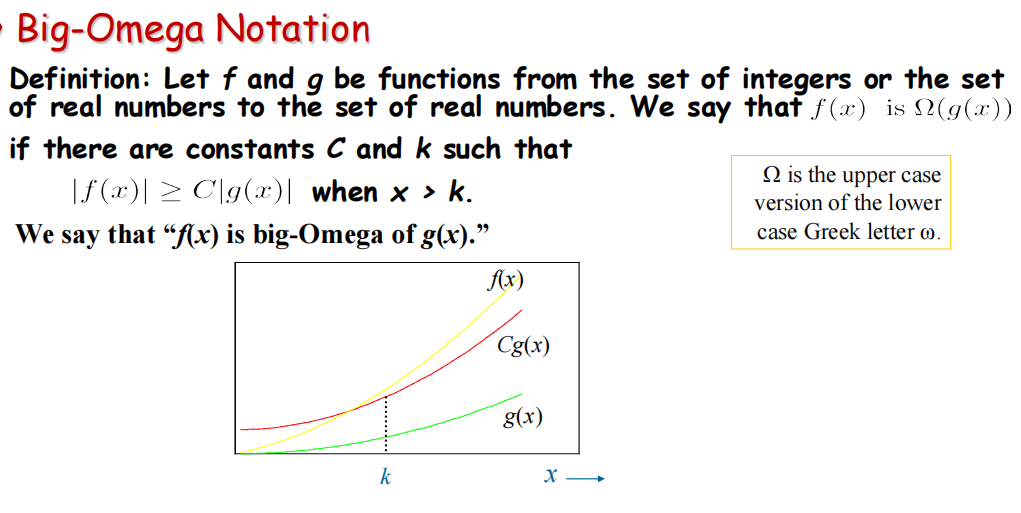




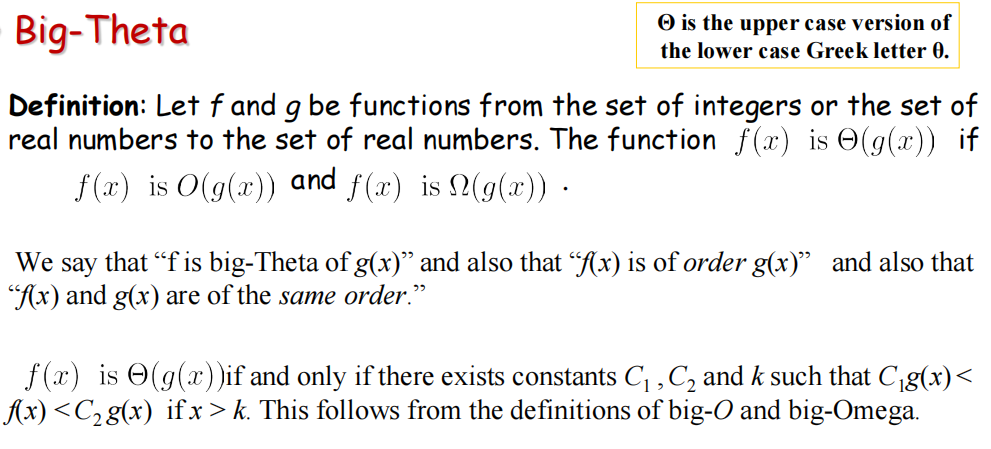


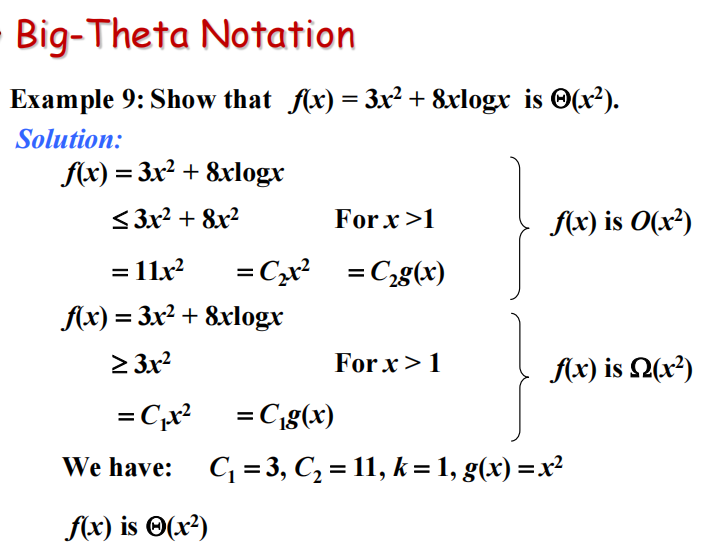


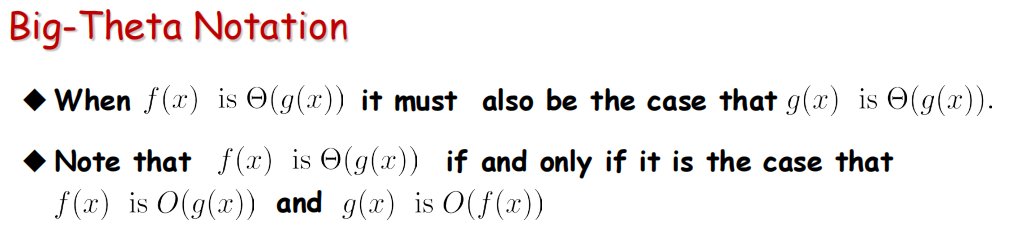


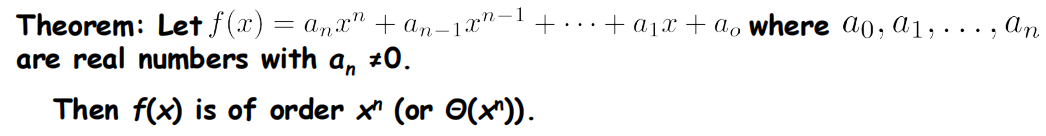


**f(x) is Ω(g(x)) if and only if g(x) is O(f(x)). This follows from the definitions.**









**（3.3）Complexity of Algorithms**

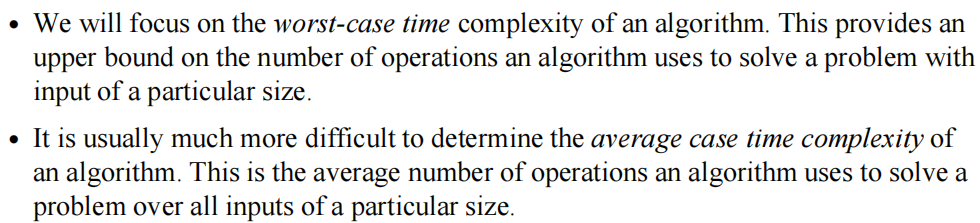
Types of Time Complexity analysis

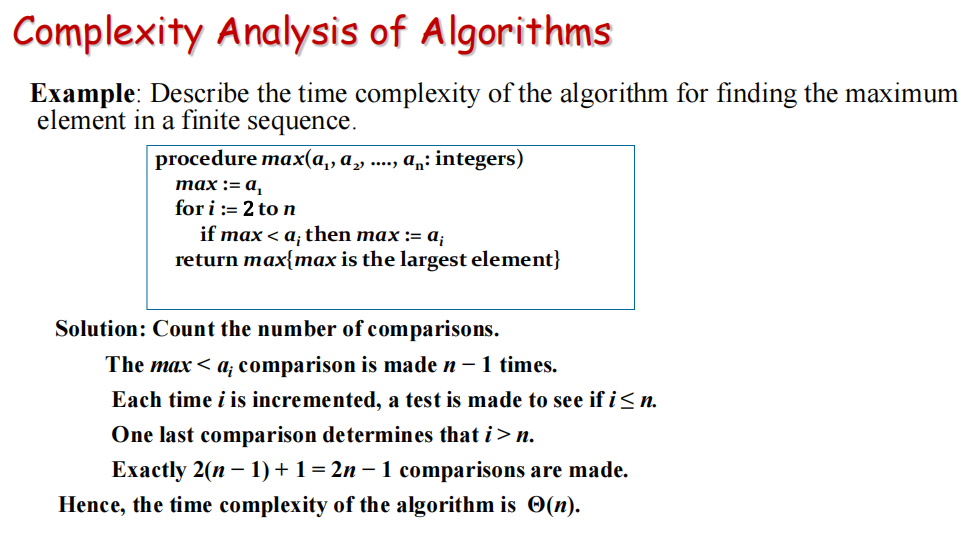
Types of Time Complexity analysis

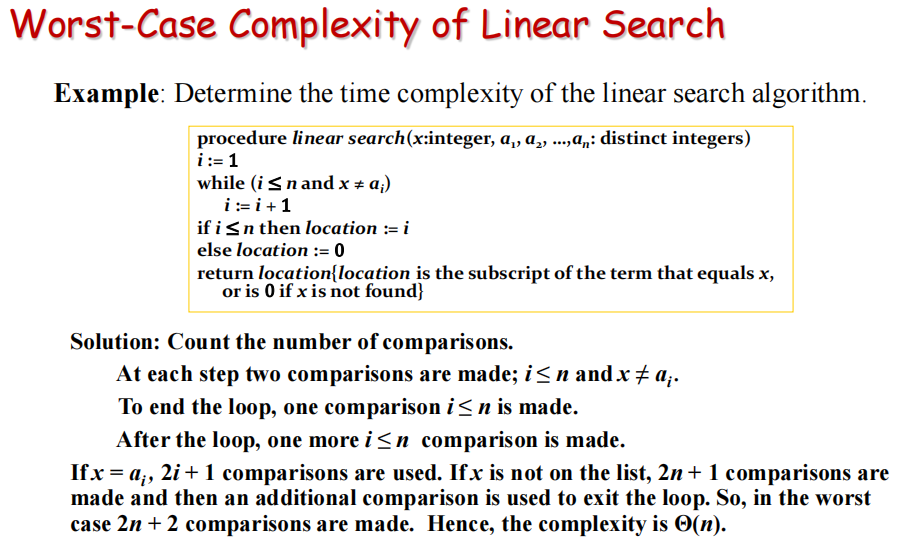
• worst-case time complexity

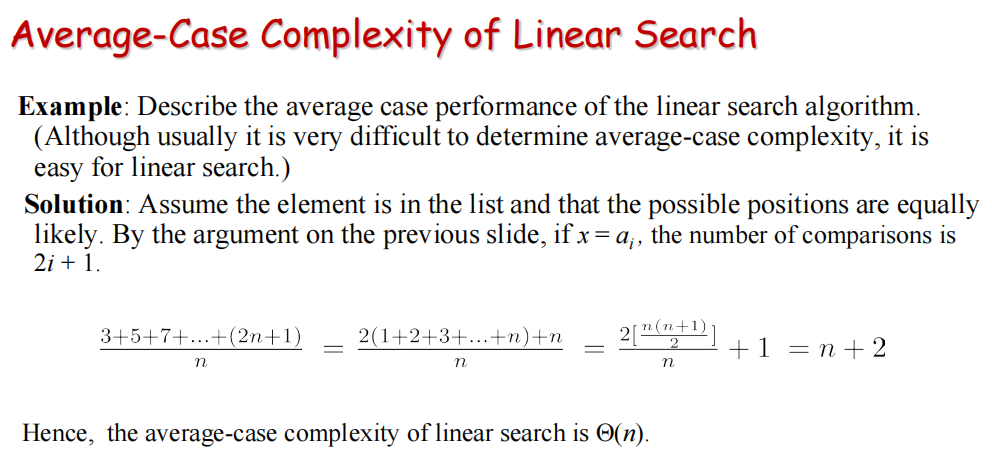
• best-case time complexity

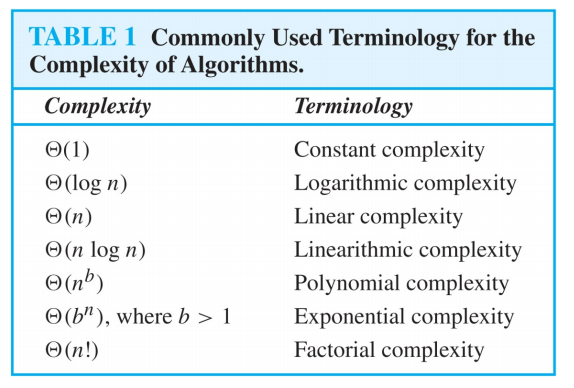
• average case time complexity











**Algorithmic Paradigms(算法范式）**