

知识总览

时间复杂度

算法效率的度量

空间复杂度

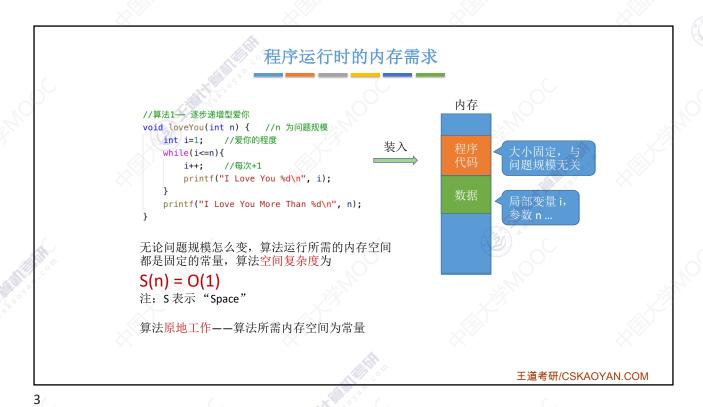
空间复杂度

空间复杂度

三道考研/CSKAOYAN.COM

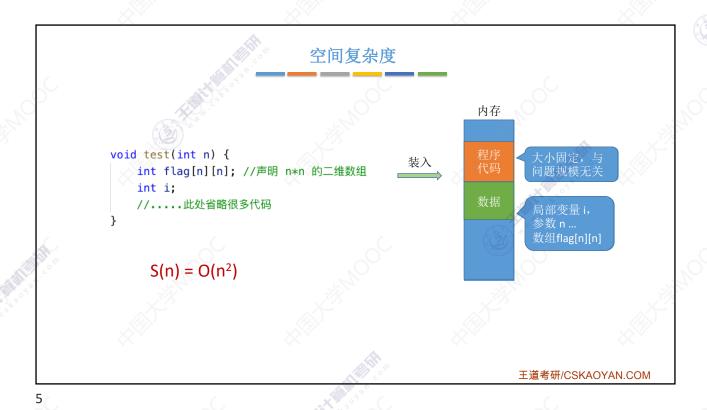
2

王道考研/cskaoyan.com

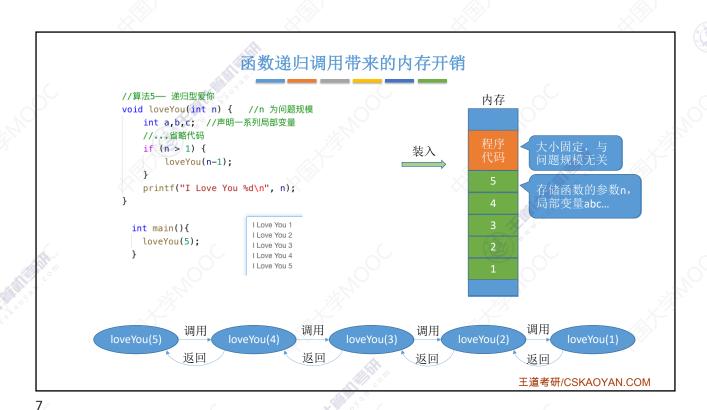


4

王道考研/cskaoyan.com

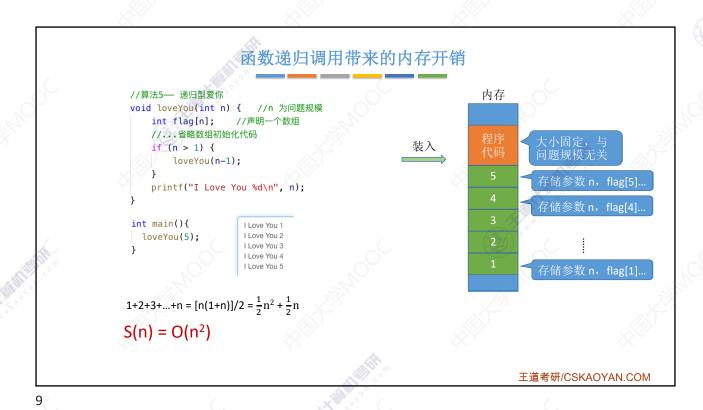


王道考岍/cskaoyan.com



函数递归调用带来的内存开销 //算法5— 递归型爱你 内存 void loveYou(int n) { //n 为问题规模 int a,b,c; //声明一系列局部变量 //...省略代码 大小固定,与 问题规模无关 if (n > 1) { 装入 代码 loveYou(n-1); 存储函数的参数n, 局部变量abc... printf("I Love You %d\n", n); int main(){ I Love You 2 loveYou(5); I Love You 3 I Love You 4 S(n) = O(n)空间复杂度 = 递归调用的深度 调用 调用 调用 loveYou(4) 返回 返回 返回 王道考研/CSKAOYAN.COM

王道考妍/cskaoyan.com



知识回顾与重要考点 ①找到所占空间大小与问题规模相关的变量 普通程序 ②分析所占空间 x 与问题规模 n 的关系 x=f(n) ③ x 的数量级 O(x) 就是算法空间复杂度 S(n) 如何计算 ①找到递归调用的深度 x 与问题规模 n 的关系 x=f(n) 递归程序 ② x 的数量级 O(x) 就是算法空间复杂度 S(n) 注: 有的算法各层函数所需存储空间不同,分析方法略有区别 空间复杂度 加法规则: O(f(n)) + O(g(n)) = O(max(f(n), g(n)))乘法规则: $O(f(n)) \times O(g(n)) = O(f(n) \times g(n))$ 常用技巧 "常对幂指阶" $O(1) < O(\log_2 n) < O(n) < O(n\log_2 n) < O(n^2) < O(n^3) < O(2^n) < O(n!) < O(n!)$ 王道考研/CSKAOYAN.COM

10

王道考妍/cskaoyan.com



王道考研/cskaoyan.com