# 时间复杂度

时间复杂度是指执行算法所需要的计算工作量。

简单来说，时间复杂度指的是语句执行次数。

计算时间复杂度的方法：

用常数1代替运行时间中的所有加法常数

修改后的运行次数函数中，只保留最高阶项

去除最高阶项的系数

按数量级递增排列，常见的时间复杂度有：

常数阶O(1),对数阶O(log2n),线性阶O(n),

线性对数阶O(nlog2n),平方阶O(n^2)，立方阶O(n^3),…，

k次方阶O(n^k),指数阶O(2^n)。

随着问题规模n的不断增大，上述时间复杂度不断增大，算法的执行效率越低。

推荐博客： <https://blog.csdn.net/zxm490484080/article/details/72210501>

# 贪心

贪心算法（英语：greedy algorithm），又称贪婪算法，是一种在每一步选择中都采取在当前状态下最好或最优（即最有利）的选择，从而希望导致结果是最好或最优的算法。比如在旅行推销员问题中，如果旅行员每次都选择最近的城市，那这就是一种贪心算法。

贪心算法在有最优子结构的问题中尤为有效。最优子结构的意思是局部最优解能决定全局最优解。简单地说，问题能够分解成子问题来解决，子问题的最优解能递推到最终问题的最优解。

贪心算法与动态规划的不同在于它对每个子问题的解决方案都做出选择，不能回退。动态规划则会保存以前的运算结果，并根据以前的结果对当前进行选择，有回退功能。

贪心法可以解决一些最优化问题，如：求图中的最小生成树、求哈夫曼编码……对于其他问题，贪心法一般不能得到我们所要求的答案。一旦一个问题可以通过贪心法来解决，那么贪心法一般是解决这个问题的最好办法。由于贪心法的高效性以及其所求得的答案比较接近最优结果，贪心法也可以用作辅助算法或者直接解决一些要求结果不特别精确的问题。

来源：wiki

大家可以先简单了解一下时间复杂度和贪心的概念。我们课上会详细讲解。如果有任何问题，可以在群里提出来。