

实验四 贪心算法实现最佳任务调度实验

一、实验原理（详细请参考课本第 16 章）

1. 活动选择问题：

对几个互相竞争的活动进行调度，它们都要求以独占的方式使用某一公共资源。而在同一时间内只有一个活动能使用这一资源。假设有一个需要使用某一资源的 n 个活动组成的集合 $S=\{a_1, a_2, a_3, \dots, a_n\}$ 。每个活动 a_i 都有一个要求使用该资源的起始时间 s_i 和一个结束时间 f_i ，且 $s_i < f_i$ 。如果选择了活动 i ，则它在半开时间区间 $[s_i, f_i)$ 内占用资源。若区间 $[s_i, f_i)$ 与区间 $[s_j, f_j)$ 不相交，则称活动 i 与活动 j 是兼容的。

活动选择问题就是要选择出一个由互不兼容的问题组成的最大子集合。

2. 贪心策略

动态规划是贪心算法的基础。

贪心算法即通过做一系列的选择来给出某一问题的最优解。对算法中的每一个决策点，做一个当时最佳的选择。

贪心算法的使用条件：贪心选择性质和最优子结构是两个关键的特点。如果我们能够证明问题具有这些性质，那么就可以设计出它的一个贪心算法。

- 贪心选择性质：一个全局最优解可以通过局部最优（贪心）选择来达到。
- 最优子结构：对一个问题来说，如果它的一个最优解包含了其子问题的最优解，则称该问题具有最优子结构。

贪心算法的基本思路：

- 建立对问题精确描述的数学模型，包括定义最优解的模型；
- 将问题分解为一系列子问题，同时定义子问题的最优解结构；
- 应用贪心原则确定每个子问题的局部最优解，并根据最优解的模型，用子问题的局部最优解堆叠出全局最优解。

二、实验要求

实现一个任务调度问题（课本 P241）：在单处理器上具有期限和惩罚的单位时间任务调度

1. 实现这个问题的贪心算法，并写出流程图或者伪代码。
2. 将每个 w_i 替换为 $\max\{W_1, W_2, \dots, W_n\} - w_i$ 运行算法、比较并分析结果。

	Task						
a_i	1	2	3	4	5	6	7
d_i	4	2	4	3	1	4	6
w_i	70	60	50	40	30	20	10

图 16-7 单处理器上带期限和惩罚的单位时间任务调度问题的一个实例

注意：当节课实验结束前找老师检查，不需提交实验报告