

第五章作业

2020/202

2020/11/189

李博云.

5. 现有一台计算机, 在某个时刻同时到达了 n 个任务。该计算机在同一时间只能处理一个任务, 每个任务都必须被不间断地得到处理。该计算机处理这 n 个任务需要的时间分别为 a_1, a_2, \dots, a_n 。将第 i 个任务在调度策略中的结束时间记为 e_i 。请设计一个贪心算法输出这 n 个任务的一个调度使得用户的平均等待时间 $\frac{1}{n} \sum e_i$ 达到最小, 并进行时间复杂性分析。

使平均等待时间最小。即处理时间短的任务优先。降低后续等待时间。所以先用一个数组, 存储输入的任务时间。然后将其从小到大排序。将排序结果存储到另一个数组中之后。用该数组计算完成时间 $e_1 = b_1$, $e_2 = e_1 + b_2$ 以及平均等待时间 $\frac{1}{n} \sum e_i$ 。

```
def rank_tasks(task)
```

```
    sorted_tasks = sorted(task) # 用python库进行升序排序
```

```
    end_times = [0] * len(sorted_tasks)
```

```
    total_end_time = 0 # 初始化和计算所用时间。
```

```
    for duration in sorted_tasks:
```

```
        total_end_time += duration # 计算总时间。
```

```
        end_times[sorted_tasks.index(duration)] = total_end_time
```

```
    average_ei = sum(end_times) / len(end_times)
```

```
    print return ei' = end_times ; ave-ei' = average-ei }
```

```
    print(f"ei = {result['ei']}")
```

```
    print(f"avg-ei = {result['avg-ei']}")
```

时间复杂度: 排序: $O(n \log n)$ 。

