

数据结构 第四次 OJ 作业

徐浩博 2020010108

- 重名剔除

本题我采用了双哈希的算法，且第一遍为正向哈希，第二遍为反向哈希，最终得到两个哈希值。我将两个值存入一个 struct 里，并且记录该字符串输入的顺序，之后根据 hash 值对 struct 数组进行排序，并且依据排序结果进行重复剔除（排序后重复的肯定相邻），再根据输入顺序排序，输出最终结果。

- 玩具

事实上，这种玩具的状态在百万级别，我们通过 dfs 搜索出所有的状态可能，然后通过已知的三种规则获得状态转移的方法。我们反向思考，从输入态转化为终态的步骤数，等价于反向转换的步骤数。因此，在状态转移图中，对于每一种状态转移，我们均一条末态指向初态的单向边。由于最终的末态均要回到 12345678 这种情况，因此这种状态可以作为超级源，我们寻找到达其他状态的最短路径。在这里，我采用了堆优化（优先队列优化）的 Dijkstra 算法。最后对于每个输出，找出它对应的状态序号，可以直接输出最短路径长度。

- 任务调度

这道题是一道比较裸的优先队列题。我们用小顶堆实现。对于每一个任务进程完成后，当优先级翻倍满足小于 2^{32} 时，我们将它修改优先级后重新加入堆里，之后每次弹出堆顶，该任务即优先级最小的任务。以此类推，直到时间用尽或堆已空。