

CLRS Ex15.4-6 实验报告

洪方舟

Student ID: 2016013259

Email: hongfz16@163.com

March 17, 2018

1. 实验目的

- a. 编写程序实现复杂度为 $O(n \lg n)$ 的算法寻找数组中最长递增子序列
- b. 加深对动态规划思想的认识

2. 实验环境

操作系统: ubuntu 16.04 LTS

处理器: Intel Core i7-7700k CPU @ 4.20GHz * 8

编程语言: C++

编译器: g++

3. 实验方法

- a. 考虑长度为 i 的递增子序列末尾数字的最小值
- b. 在 dp 数组中第 i 个位置存储长度为 i 的递增子序列末尾数字的最小值, 每次更新 dp 数组的时候二分查找不大于待插入数字的最后一个位置
- c. 当完成遍历后 dp 数组的长度就是最长递增子序列的长度, 通过回退查找可以构造出一个最长

递增子序列

- d. 使用 $O(n^2)$ 的算法来保证本算法的正确性
- e. 使用Qt制作方便测试的图形界面

4. 实验分析

算法中总共只有一个循环，复杂度为 $O(n)$ ，在每一个循环中，需要去 dp 数组中查询插入位置，该插入位置 i 应满足 $dp[i] \leq NumberToInsert < dp[i+1]$ ，使用二分查找的复杂度为 $O(\lg n)$ ；综上所述算法总复杂度为 $O(n \lg n)$ ；

5. 实验总结

- a. 通过两种算法的验证，可知本算法具有一定的正确性
- b. 对于 dp 数组的不同的理解方法可以带来不同的算法复杂度

6. 源代码及可执行文件说明

- a. 源代码存放在`/src/LISSrc`
- b. 可执行文件存放在`/bin/LIS`
- c. 打开可执行文件后是用户界面，可以在输入框内输入一串用空格隔开的数组（若输入不合法将会给出提示），点击Submit!按钮后在下方将最长递增序列标红显示，并给出最大长度和运算时间