



中国石油大学 (华东)  
CHINA UNIVERSITY OF PETROLEUM

# 本科实习手册

实习名称 数据结构与算法实习

实习单位

实习时间 2020.6.21 至 2020.6.27

学 院 计算机科学与技术学院

专业班级 计科 1802

姓 名 张世琛

学 号 1804030401

指导教师 孙歧峰

2020 年 6 月

## 填写说明

一、参加实习的学生必须每天对实习情况做日记。实习日记应准确的反映实习当天的情况，包括实习时间、地点、实习方式和实习内容、遇到的问题及收获心得等。

二、实习结束之前，学生必须认真撰写实习报告。通过撰写报告，系统地回顾和总结实习的全过程，将实践性教学的感性认识升华到一定的理论程度，从而提高实习教学效果。

三、实习报告的撰写应符合实习大纲的要求。实习报告的质量反映了实习的质量，它是实习成绩评定的主要依据之一。实习报告需经指导教师审阅并签署意见。

四、实习日记和实习报告要求手写，条理清晰、内容详实、层次清楚、字迹工整。

五、实习结束后, 学生将实习手册上交开课学院存档。

实习地点：家中

实习方式：做数据结构与算法题目

实习内容：数据范围查询一题

遇到的问题及收获心得：

① 数据范围为  $1 \times 10^8$ , 256MB 内存则完全能开  $1 \times 10^8$  的 int 数组, 而且不能超过  $2^{31}-1$  的

② 利用前缀和来优化加速询问区间  $[a, b]$   
 $sum[i] = \sum_{j=1}^i a[j]$ ,  $a[i]$  代表坐标为  $i$  的个数,  
 $sum[i]$  代表区间  $[1, i]$  点的个数

③ 对于每次询问,  $ans = sum[R] - sum[L-1]$ ,  $O(1)$  得出  $[L, R]$  区间点的个数.

实习地点：家中

实习方式：做数据结构与算法题目

实习内容：灯塔 (LightHouse) - 题.

遇到的问题及收获心得：

① 先按照  $x$  从小到大排序,  $x$  相同按  $y$  从小到大排序  
 一对灯塔相互照到  $\begin{bmatrix} \cdot & (j) \\ i & (i) \end{bmatrix}$  ( $y_i < y_j$ ) 即排序后  $y$  的顺序对数

② 求顺序对数参考逆序对数, 利用归并排序  
 $\begin{matrix} L & i & mid & mid+1 & j & R \end{matrix}$  若  $y_i < y_j$  则这一对的贡献为  $r-j+1$ , 累加到答案即可

③ 时间复杂度:  $O(n \log n)$  空间复杂度:  $O(n)$

实习地点：家中

实习方式：做数据结构与算法题目。

实习内容：列车调度 (Train) - 题

遇到的问题及收获心得：

- ① 这道题目是最基本栈的应用。1-n 的顺序入栈，<sup>每输入一个 x</sup>若当前栈顶的值 <sup>并标记该操作为进栈 (1)</sup>小于 x，则进栈元素，直到栈顶元素为 x，在这个过程中若栈的大小大于 m，则输出 No，结束程序。
- (2) 当前栈顶元素大于 x，根据栈的性质先进后出，则 x 在栈中某一位。若将 x 取出，则必顺取出 x 上面元素。这显然不可以。输出 No，结束程序。
- (3) 若程序执行到此，则栈顶元素必为 x，则弹栈，标记此步骤为弹栈 (0)。(4) 最后程序没有 return 0，则输出标记，输出 "push"。

实习地点：家中

实习方式：做数据结构与算法题目。

实习内容：旅行商 (TSP) - 题

遇到的问题及收获心得：(DAG 图)

由题意得，此题为有向无环图。基本的思路为拓扑排序 + dp 求最长路径。为什么这样的思路呢？首先 dp 要考虑，该状态从哪一点转移过来，而由拓扑排序的性质得，拓扑序中的边总是由前面的点指向后面的点，也就是说该点只能从前面的点转移过来。这与 dp 思想不谋而和（也就是解决 dp 中无后效性的问题）。而至于拓扑排序则是利用队列，先将入度为零的点入队，从队首取元素，将其关联的边删去，若某点入度为零，则入队，由此往返执行，则得出拓扑序列。



实习地点:家中

实习方式:做数据结构题目

实习内容:(任务调度(Schedule))-题

遇到的问题及收获心得:

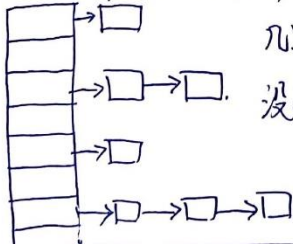
此题为堆的一个应用. 先前将  $n$  个数插入, 向上调整, 若  $\text{heap}[i] < \text{h}[i/2]$  交换, 否则退出.  $m$  次操作, 删除操作  $\text{heap}[1] = \text{heap}[n--]$  下调:  $i = i * 2$ , 比较左右儿子, 若右儿子小则  $i++$ , 若  $\text{heap}[i] < \text{heap}[i/2]$  交换, 否则 break. 若以前堆顶  $\text{key} \leq 2^{32}$  再次入堆, 向上调整. 而此题利用堆排有几个明显优势, 论时间复杂度的稳定性比快速排序更优, 而论排序中额外空间比归并排序更优. 归并排序需额外  $O(n)$  的空间作为辅助.

实习地点:家中

实习方式:做数据结构题目

实习内容:做(重名剔除)-题 (Deduplicate)

此题为 Hash 题目, 首先无论我变了多少模数, 还有进制 (我用进制 Hash) 都过不了此题 (可能有 Hash 冲突), 而最后用了开散 (拉链法) 列表法. 用链表维护 hash 值相同的数据, 每个点记录出现了几次, 若出现第二次, 则输出, 若在此 hash 值没有, 则加上此结点. 时间复杂度  $O(n)$  空间复杂度  $O(n)$ , 解决 hash 冲突的方法还有开放定址法 (线性探测, 平方探测).



实习地点:家中

实习方式:做题结构题目

实习内容:平均气温 (Temperature)

遇到的问题及收获心得:

KD-tree: KD-tree 每个结点, 存储了一个  $k$  维空间域和一个  $k$  维空间坐标, 而构建树时按照每一维的中位数, 随着树的深度增加循环地选取坐标轴作为分割超平面的法向量. 每次均为当前点集中, 选中某一维坐标的中位数的点作为切分点, 切分点作为父结点, 左右两侧为划分作为左右子树. 查询, 如果要查询的区域完全包含于目标矩形区域, 那就跟线段树查询一样直接返回区域和就好了. 否则接下来要查询的区域与目标区域有交集, 就往下查, 否则不查.

实习日记

年 月 日

实习日记

年 月 日

实习日记

年 月 日

实习日记

年 月 日



实习日记		年	月	日
实习日记		年	月	日

实习日记	年 月 日
实习日记	年 月 日

实习日记		年	月	日
实习日记		年	月	日

实习日记

年 月 日

实习日记

年 月 日



实习日记		年	月	日
实习日记		年	月	日

实习日记		年	月	日
实习日记		年	月	日

## 实习报告

实习报告应包含本次实习的基本情况介绍、实习目的、实习内容、实习的收获（学到的知识、掌握的技能、心得体会等）

本次实习从2020年6月21日到2020年6月27日,总共12道题目。《数据结构与算法》是计算机科学与技术专业中是一门实践性较强的专业基础课,上机实习是对学生的一种全面综合训练,是与课堂听讲、自习和练习相辅相成的必不可少的一个教学环节。实习着眼于原理与应用的结合,使学生学会把学到的知识用于解决实际问题,起到深化理解和灵活掌握教学内容的目的。同时

实习内容包括简要描述题目要求,对问题的描述应避免开算法及所涉及的数据类型,只要对所完成的任务做出明确的陈述,例如输入数据的类型、值的范围,以及输入的形式、输出数据的类型、值的范围及输出的形式。②选定数据结构写出算法,首先描述算法的基本思想,然后进行算法细化,再对所设计的算法的时间复杂性、空间复杂性进行简单分析。

实习的内容中有一道平均气温,是利用KD-tree的课题,就是询问一个区域的平均气温,就是为了得到一个区域气温总和和测试点的个数;将气温总和sum除以测试点的个数Cnt就是所要的答案,接下来我将写出我对KD-tree的理解;

kD-tree 中的一个结点, 存储一个 k 维空间域和一个 k 维空间的点坐标, 它的节点存储方式和 Splay Tree 有异曲同工之妙, 可以把 k 维空间域当作子树信息维护: k 维空间域就是子树内<sup>所有</sup>结点的坐标范围

```
Struct Point {
```

```
    int x[2]; // 坐标点
```

```
    int mi[2], mx[2] // 坐标域的范围
```

```
    int sum, size; // 气温之和, 气象点个数
```

```
    int v; // 该点气温
```

```
    int l, r; // 左右儿子
```

```
}
```

KD-Tree - 一颗平衡二叉树, kD 即为 k-Dimension, 每个节点即为一个 k 维的点, 每个叶子结点可以想象为一个分割超平面, 用于垂直坐标轴的超平面将空间分为两个部分, 这样递归从根结点不断划分, 直到没有点为止。

经典构造 kD-tree 的规则如下。

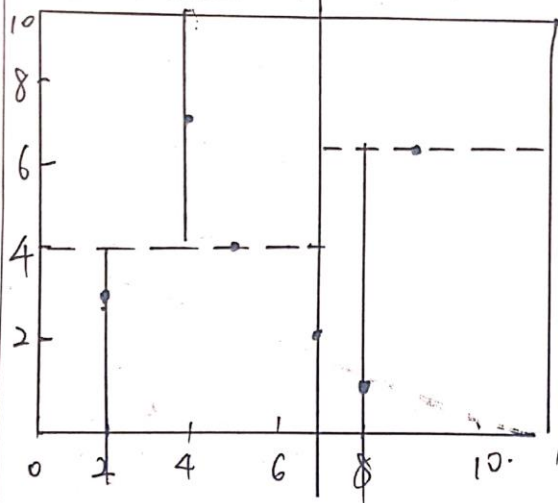
① 随着深度增加, 循环选取坐标轴, 作为分割超平面的法向量。对于 2D tree 树来说, 根结点选取 x 轴, 根结点的孙子选取 y 轴, 根结点的孙子选取 x 轴, 这样循环下去。

② 每次均为当前点集中, 选中某一维坐标的中位数作为切分点, 切分点作为父结点, 左右两侧为划分的左右两子树

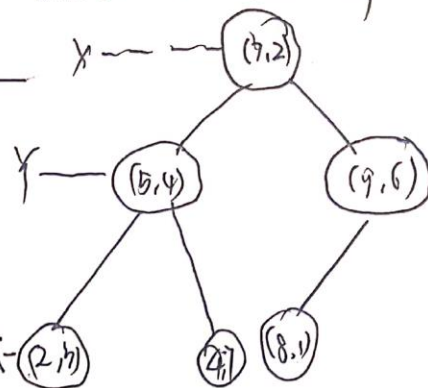
建立 kD-tree 的时间复杂度为  $O(kn \log n)$  ( $n$  为点数,  $k$  为维数)



例如：存在 6 个点 (2,3) (5,4) (9,6) (4,7) (8,1) (7,2)



实线为x轴，虚线为y轴



建树 `int build (int l, int r, int k) {`

`if (l > r) return 0;`

`int mid = (l+r) >> 1; k=k`

~~std::nth\_element~~ `nth_element (p+l, p+mid, p+r+1)`

~~k~~ `tree[mid] = p[mid]`

`tree[mid].l = build (l, mid-1, k+1);`

`tree[mid].r = build (mid+1, r, k+1);`

`update (mid);`

`return mid;`

`}`

② 用 `nth_element` 函数求中位数，应自己手写，测试平台不提供 STL，利用中位数使左右两棵子树尽可能平衡。

回顾此课程设计,至今仍感慨颇多,从理论到实践这段日子里,可以说得苦多于甜,但是在这过程中可以学到很多很多东西,同时不仅可以巩固以前所学的知识而且学到了很多书本上没有学过的知识。

借用德国诗人歌德的诗剧《浮士德》中一句话,  
“一切理论都是灰色的,而生命之树常青”

2020 年 6 月 27 日

## 指导教师评语

(请结合实习大纲要求, 从学生实习态度与表现、实习任务完成情况、实习报告撰写等方面进行客观评价)

指导教师签字: \_\_\_\_\_

年      月      日