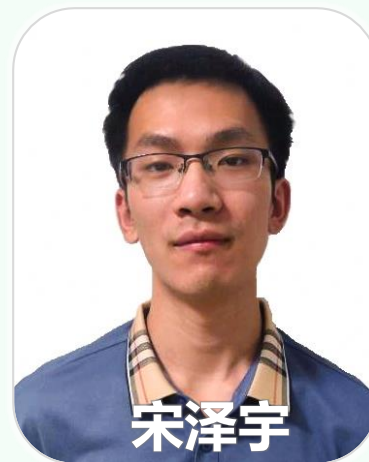


“数据结构” 课程大纲

2022秋季学期

清华大学 · 计算机科学与技术系

教学团队



考评环节

课号	30240184	00240074	视疫情态势，可能会做进一步调整
编程作业PA 专题实验LAB	≤ 3 x(11+3)题/4次	≤ 4 x(9+4)题/4次	通过网络学堂发布的 注册码 ，在OJ上 加入课堂 ；请尽早阅读OJ首页上的 帮助文档 ，了解作业、实验 形式 、相关 纪律 及重要的 基本技术 ；充分利用 习题课 、 答疑 环节获取帮助
问题集 Problem Set	0		课程内容的验证、补充与探究，随讲授进度分批次发布鼓励 独立思考 ，可以充分地 自由 研讨
考试/Exam	58 考试周，期末	48 最后一周随堂，期末	全闭卷 笔试 基本知识点的 充分 掌握，基本方法和技巧的 灵活 运用
加分/Bonus	5		勘误 + 测例设计 + 作业创意 + 进步幅度
总评/Grading	按百分制累计成绩确定 排名 ，参照近年来的 分布 划分等级		

实践环节

PA: 针对实际问题，综合运用讲授内容

LAB: 深入了解测试与调试的过程，掌握相关方法与技巧
深入了解同类数据结构或算法的特点，学会灵活运用

你以为程序员经常思考：



其实他们可能常常在想：



< DSA-CST-19秋(287) ...



突然有一点好奇，PA 每次截止时间 21:55 是怎么选出来的



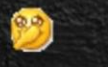
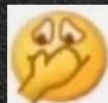
随手取的时间吗



(健康作息下的结束工作准备睡觉的时间 x

2019年12月28日 晚上18:20

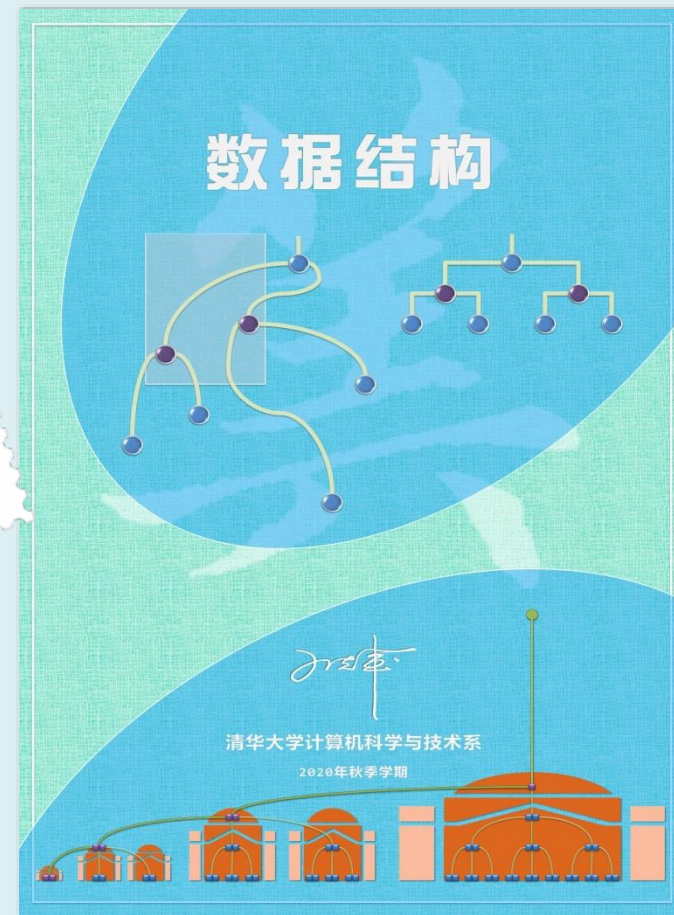
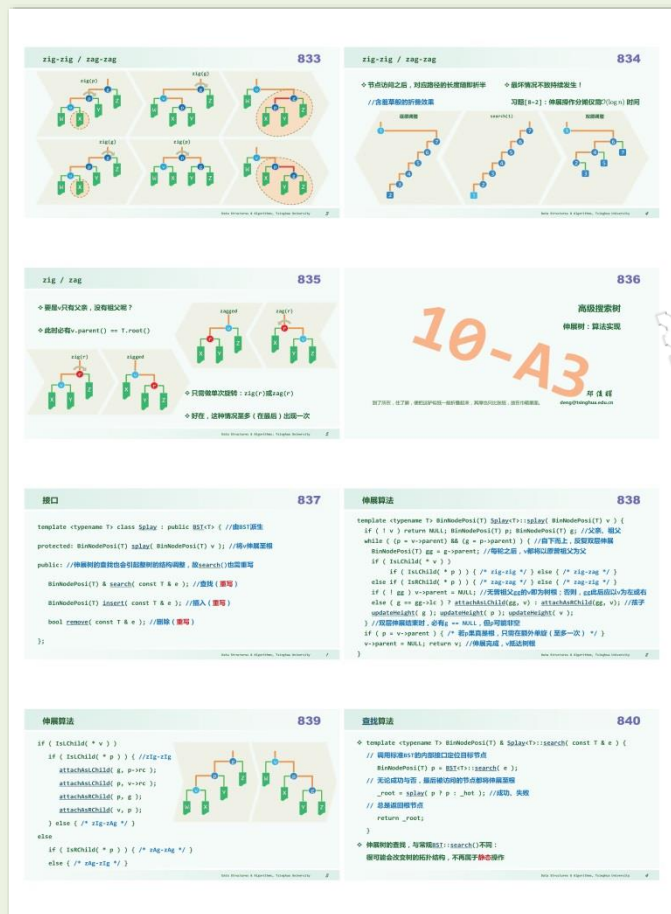
谐音，我要呜呜



基础知识

- ❖ DSA涉及多个学科，但**并不意味着**必须逐一**精通**，常用部分只是其中一个不大的**子集**
- ❖ **C/C++语言**： 对象/属性方法、类/构造/析构、继承、重载、重写、虚方法、模板
- ❖ **离散数学**： 集合、偏序集、良序、数学归纳法、
级数、递归/递推、排列/组合、Stirling逼近、Fib数、Catalan数
- ❖ **概率与统计**： 概率、随机分布、数学期望、期望值的线性率、几何分布、二项式分布
- ❖ ...
- ❖ 关于数学，我们会尽可能使用**初等**的方法

电子阅读版：按章节独立PDF文档，提供超链接索引



纸质版：多种板式，目录统一；彩色、黑白打印两相宜

学堂在线: <https://www.xuetangx.com/search?query=邓俊辉>

- ✓ **数据结构**（上、下）：14章100余节，共500余段视频
- ✓ **计算几何**：8章88节，共400余段视频
- ✓ 覆盖校内课堂的**大部分**讲授内容，是重要的学习辅助渠道
- ✓ 与校内课堂所讲授内容的对应关系，详见打印版讲义目录中的**标注**
- ✓ 视频之间有大量Quiz，虽难度不大，仍**有助于**确认自学效果

数据结构 (C++) : <https://dsa.cs.tsinghua.edu.cn/~deng/ds/dsacpp/>

- ✓ 内容是讲义的**子集**，讲解详细但欠深入；校内学生**不建议**购买，可主要依靠讲义
- ✓ 习题解析：针对讲授内容落实巩固，基本部分针对考试，进阶部分着眼拓展，**值得**购买
- ✓ 出版十年来已发现大量错误，阅读前请下载**勘误表**，并对照更正
- ✓ 如有发现新的错误，欢迎报告到网络学堂讨论区的**勘误报告**专区



本课程相关的多数**学习资料**，均可经由**教材主页**上对应的链接，从清华网盘**自由下载**

样卷 + 真题

3. 判断 (请使用 '0' 和 'X')

- ☐ 在某节点被删除后AVL树的高度即使下降
- ☐ 有向图经DFS后若共有k条边被标记为BACK
- ☐ 对于同一无向图, 起始于顶点v的DFS遍历
- ☐ 采用单向平方策略的散列表, 只要长度
- ☐ PFS过程中, 尽管每一步迭代都可能多
- ☐ 相对于KMP算法而言, BM算法更适合
- ☐ 在存有n个词条的跳转表中, 各词条高度
- ☐ 红黑树的插入或删除操作, 都有可能
- ☐ 将{0, 1, 2, ..., 2018}插入一棵空的伸展
- ☐ 在插入操作后若红黑树黑高度增加,
- ☐ 若输入序列包含 $\Omega(n^2)$ 个逆序对, 则
- ☐ 采用12-C节中介绍的任何一种增量

6. KMP

所谓的斐波那契串 (Fibonacci Strings), 对于 $k \geq 2$, 有 $\phi_k = \phi_{k-1}\phi_{k-2}$, 比如: $\phi_2 =$

1) 以下考查KMP算法的改进版。试列出 ϕ_k

j	0	1	2	3	4	5
$\phi[j]$						
improved_next[j]						

2) 若 $|\phi| \geq 2$, 则将 ϕ 末尾的两个字符翻转

试证明: $\forall k \geq 2, \phi_{k-2}\phi_{k-1} = \phi_k$

3) 试证明: 若以 ϕ_k 作为模式串, 文本串的

4) 试证明, 对于任何模式串P, 文本串

8. Reconstruction

// 由先序、中序遍历序列, 重构一棵由n个节点
BinNodePosi<T> reconstruct(T pre[], T
if (n < 1) return NULL;
k = search(pre[0], in, n); // loca
return new BinNode(
pre[a], //key
reconstruct(
reconstruct(
);

A) 试补充递归调用处缺失的参数; B) 设每次

9. Bitmap & QuadTree

各像素或黑或白的图像, 也称作位图; 正

```
struct QuadNode {  
    bool black; //若是叶节点, 则非黑即  
    QuadNode* child[4]; //按象限排序  
    QuadNode( bool b ) //按指定颜色b  
    { black = b; for ( int i = 0
```

};

(子) 位图与 (子) 四叉树的对应规则为:

- 如果位图中的所有像素同色, 则对应于

- 否则, 将该位图平均划分为4块子位图

A) 试画出如下中间那幅位图所对应的四

4. 计算 (保留推导过程, 包括图、表, 这

设整数e独立且均匀地取自[0, 25], 现通过

k	0	1
A[k]	1	3

试分别计算其在失败情况下的平均查找长度

5. 证明 (请同时给出示意图)

在由n个节点构成的二叉树中, 任意节点v

试证明: 若二叉树的先序遍历序列为 v_0, v_1, \dots, v_{n-1}

$$\sum_{k=0}^{n-1} \|v_k v_{(k+1) \bmod n}\| = \|v_0 v_1\| +$$

姓名: 2 0 1 班

7. Relayed Search 4 + 4

我们知道, 插值查找、二分查找、顺序查找分别适用于大、中、小规模的数据。当有序向量很长时, 我们可以依次使用它们做接力式的查找。若在某系统中经测量确认, 三种算法的时间复杂度系数约为1280 : 64 : 1, 试估算出分别应在查找范围缩小到多大时切换算法 (忽略复杂度的低次项、算法切换过程的时间消耗等因素)。

8. Failstone 2 + 6

```
1 void Failstone( int p, int q ) { //0 < p < q  
2     while ( p != q )  
3         if ( p > q ) { int r = q; q = p; p = r; }  
4         else { ( q & 1 ) ? p <= 1 : q >= 1; }  
5 }
```

1) 考察如上算法, 对于Failstone(5, 13), 试列出每次执行到第2行(循环入口)时p和q的数值(退出后留空):

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p	5														
q	13														

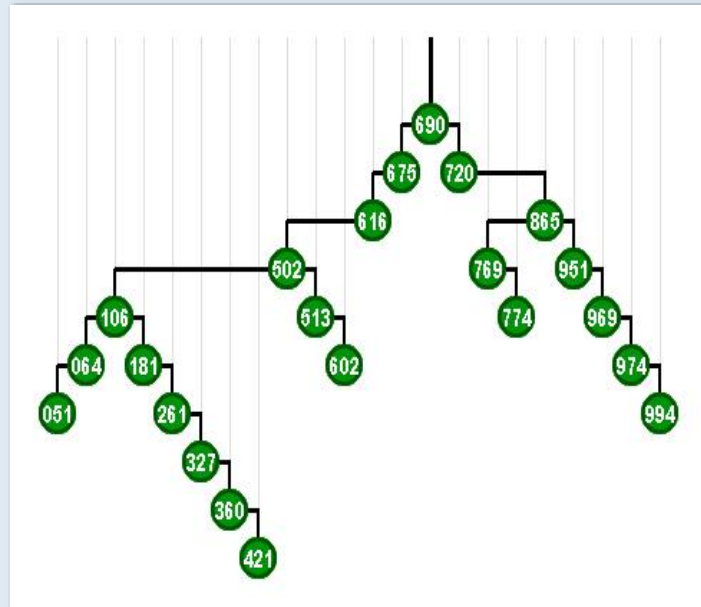
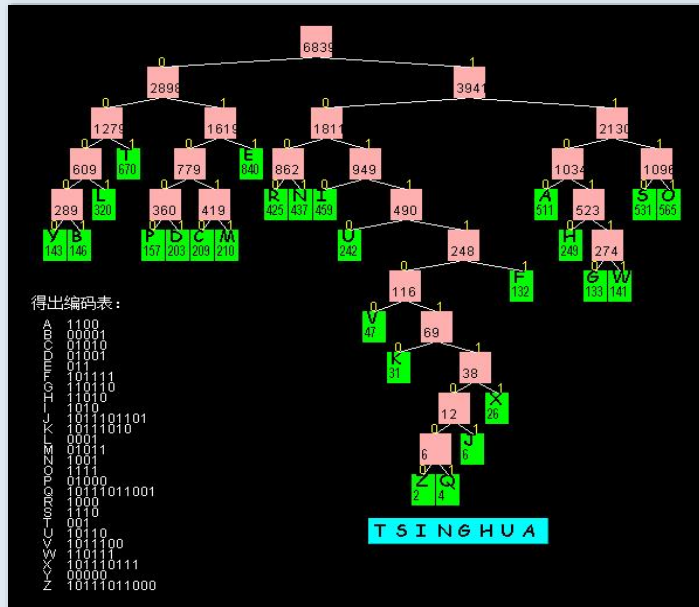
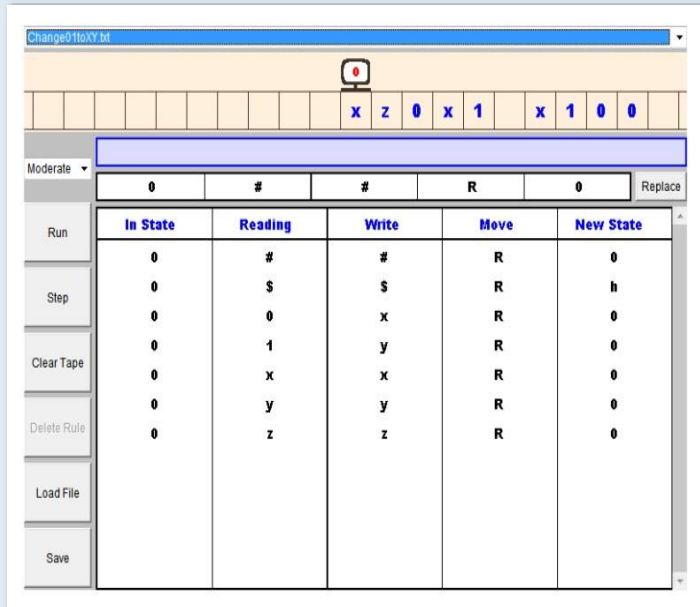
2) 试证明, 该算法必然终止 (不考虑整数溢出等因素)。



本课程相关的多数学习资料, 均可经由[教材主页](#)上对应的链接, 从清华网盘自由下载

教学演示 (Applet版)

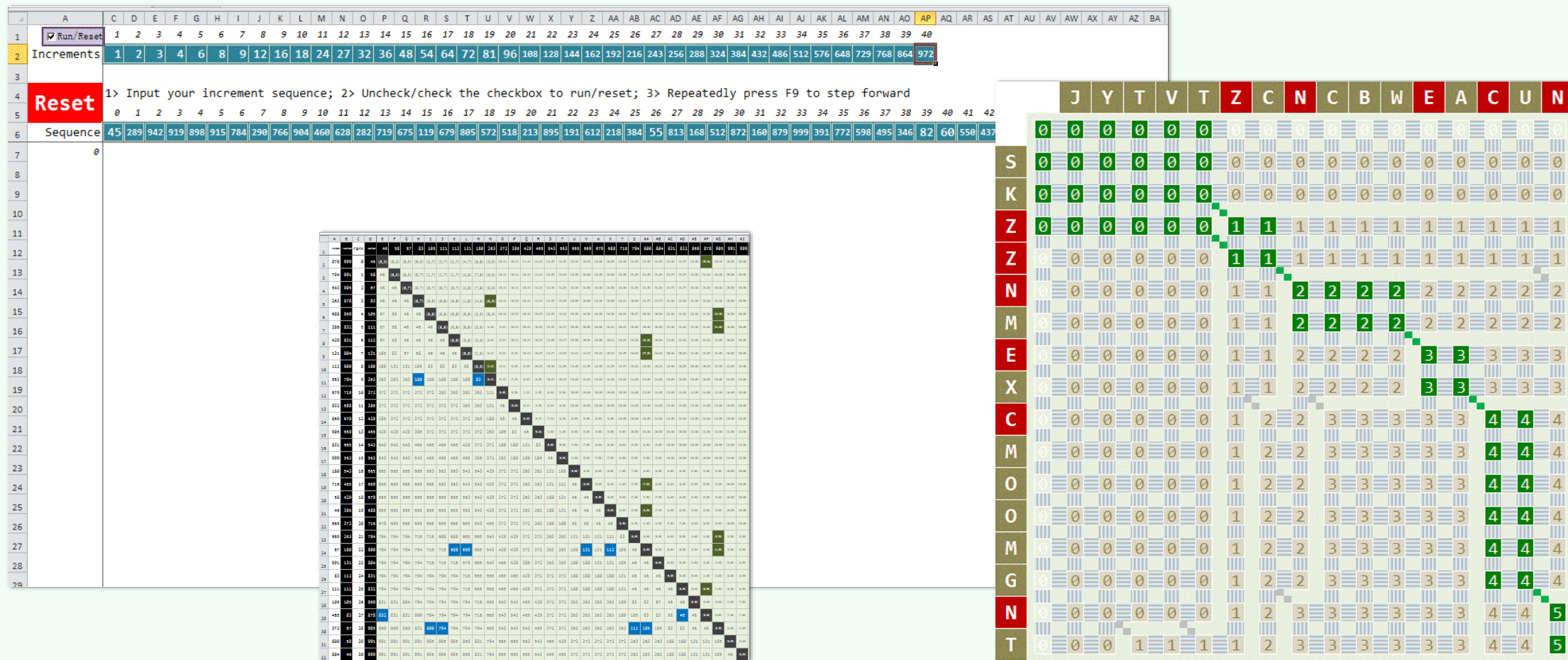
- ✓ 针对经典的数据结构与算法，通过网络学堂打包发布
- ✓ 多为Java Applet形式，可借助AppletViewer.exe播放（建议将index.applet与之绑定）



- ✓ 在老式浏览器中，也可以直接访问：<https://DSA.cs.tsinghua.edu.cn/~deng/ds/DEMO>

教学演示 (Excel版)

- ✓ 通过网络学堂打包发布：用Excel打开后，随机或手工设置输入，即可反复按F9键步进



示例代码

- ✓ Visual Studio 2019格式的60+个工程（与其他环境**未必**兼容）
- ✓ 多通过**命令行**设置输入，可直接编译执行
- ✓ **声明**后可用于PA，对可能含有的**BUG**自己负责

```
Searching for 0277 ... Not found
Inserting < 0090, W> ... Done
class Skiplist<int,char>[2291816]*12*123:
class Quadlist<struct Entry<int,char> >[3137904]*001: -----
class Quadlist<struct Entry<int,char> >[3137856]*001: -----
class Quadlist<struct Entry<int,char> >[3137808]*001: -----
class Quadlist<struct Entry<int,char> >[3137760]*001: -----
class Quadlist<struct Entry<int,char> >[3137712]*002: -----
class Quadlist<struct Entry<int,char> >[3137664]*005: -----
class Quadlist<struct Entry<int,char> >[3137616]*005: -----
class Quadlist<struct Entry<int,char> >[2955464]*006: -----
class Quadlist<struct Entry<int,char> >[2955416]*016: -----
class Quadlist<struct Entry<int,char> >[2955320]*030: -----
class Quadlist<struct Entry<int,char> >[2954248]*061: -< 0002: K>-----< 0007: M>--< 0010: M>-----< 0016: I>---
class Quadlist<struct Entry<int,char> >[2954032]*123: -< 0002: K>--< 0003: I>--< 0004: F>--< 0007: Q>--< 0007: M>--< 0010: M>-----< 0016: I>---< 0016: I>---
```

```
Searching for 1455 ... Not found
Removing 3314 ... Not exists
Searching for 0612 ... Not found
Inserting 1783 ... Done
class BTree<int>[4128208]*1:
1783 *>└─ 1783
```

```
Searching for 2822 ... Not found
Searching for 1668 ... Not found
Searching for 3469 ... Not found
Inserting 3644 ... Done
class BTree<int>[4128208]*2:
3644 *>└─ 3644
1783 *>└─ 1783
```

```
Inserting 0178 ... Done
class BTree<int>[4128208]*3:
3644 *>└─ 3644
1783 *>└─ 1783
0178 *>└─ 0178
```

```
Inserting 2307 ... Done
class BTree<int>[4128208]*4:
3644 *>└─ 3644
2307 *>└─ 2307
1783 *>└─ 1783
0178 *>└─ 0178
```

```
Removing 1957 ... Not exists < 0016: I>---
Searching for 0190 ... Not found < 0016: I>---
Inserting 1798 ... Done < 0016: I>---
class BTree<int>[4128208]*5: < 0016: I>---
3644 *>└─ 3644 < 0016: I>---
2307 *>└─ 2307 < 0016: I>---
1798 *>└─ 1798 < 0012: H>---< 0012: A>---< 0016: I>---
1783 *>└─ 1783
0178 *>└─ 0178
```

```
Inserting 1839 ... Done
class BTree<int>[4128208]*6:
```

Online Judge (OJ)

<https://DSA.cs.tsinghua.edu.cn/OJ>

Tsinghua Online Judge
Department of Computer Science and Technology, Tsinghua University

Welcome!
Quick [guide](#), [video guide](#) and PA [handbook](#) are available.

49,956 registered users, 8 active courses, and 140 closed courses.
64 test runs in the last 24 hours.
323 test runs during last week.
1,948,465 test runs since 09/01/2012.

New here? [CREATE AN ACCOUNT](#)

E-Mail
Username
Password
☐ Remember E-Mail
[Sign In](#)

MOOC数据结构(上)(自主模式)
Programming Assignment #1(2026-08-01 07:59:00)

Score	范围查询(Range)	祖玛(Zuma)	灯塔(LightHouse)
100	55	10	10
90	10	5	5
80	5	2	2
70	2	1	1
60	1	0	0
50	10	0	0
40	2	0	0
30	1	0	0
20	1	0	0
10	1	0	0
0	1	0	0



Join Private Course

Code:

[OK](#) [Close](#)



先用email注册加入平台，再从网络学堂发放的邀请码加入课堂
务必认真阅读首页上的帮助文档

- ✓ 须独立完成，否则后果严重：**双方同论**，不作区分；范围覆盖**同一课程**历届的所有学生
- ✓ 查重范围及判定依据 = 代码及文档，**雷同度**
- ✓ 流程 = 查重系统提供名单 + 人工核验 + 查看**HonorCode** + 初步判罚 + 复议 + 最终判罚
- ✓ 标准、经典算法**除外**：二分查找、KMP、Dijkstra、快速排序、DFS/BFS ...
- ✓ 什么**不可以**交流？源代码，或可导致雷同的伪代码；实验报告等参与考核的文档
- ✓ 什么**可以**交流？
 - 题意理解、解题思路（动口不动手，用耳用心不用笔纸）
 - 算法、数据结构的设计与选用方案
 - 测试用例（欢迎设计，并随同文档提交到网络学堂答疑区）
- ✓ 尺度拿捏不准？涉及的人员、文献、资源等，须在**HonorCode**中标注
只要注明得客观、准确、完整，就至少不会倒扣

习题课

- 系统讲解作业涉及的一般性技巧（尤其是关于测试、调试的技巧），以及各题的思路及要点
- 预先**录制**，录像及讲稿通过网络学堂陆续发布

网络学堂

- 在**讨论版**研讨一般的有共性的问题
- 在**答疑区**提出个性化的问题，以及提交涉及个人的材料
- 过于具体的技术细节，建议通过**答疑**环节来交流

个别答疑

- 主要针对你在完成**作业**过程中遇到的技术问题
- 线上或线下地**实时**交流
- 具体安排详见网络学堂的各次通知

交流原则

- 交流之前，需掌握课堂讲授的**基本**内容，已核对过**勘误表**，并已做过足够的独立**尝试**
- 交流过程中，尽可能**规范、简洁、全面**地描述你的问题



A
扫码关注
THU数据结构

B
加入/切换课堂
`@class=DENG`

C
此后输入的文本
均作为弹幕发送

X
绑定id
`@id=2021xxxxxx`

