

“数据结构” 课程大纲

2024春季学期

清华大学 · 计算机科学与技术系

考评环节：30240184

编程作业PA 专题实验LAB	≤ 3 x11题/4次 x3题/3次	通过网络学堂发布的 注册码 ，在OJ上加入课堂 黑盒 、 白盒 测试兼顾； 九成测 按次数罚分，并 跨学期 累计 尽早阅读 帮助文档 ；充分利用 习题课 、 答疑 获取帮助
考试 Exam	58 期末考试周	全闭卷 笔试，针对课上讲授内容 基本知识点的 充分掌握 ，基本方法和技巧的 灵活运用
问题集 Problem Set	0	讲授内容的验证、补充与探究，随进度分批次发布 鼓励 独立思考 ，可以充分地 自由 研讨
总评 Grading	按以上累积成绩确定 排名 ，参照近年来的 分布 划分等级；适量计入 退课 人数	

能否选修 + 如何学习

❖ 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**

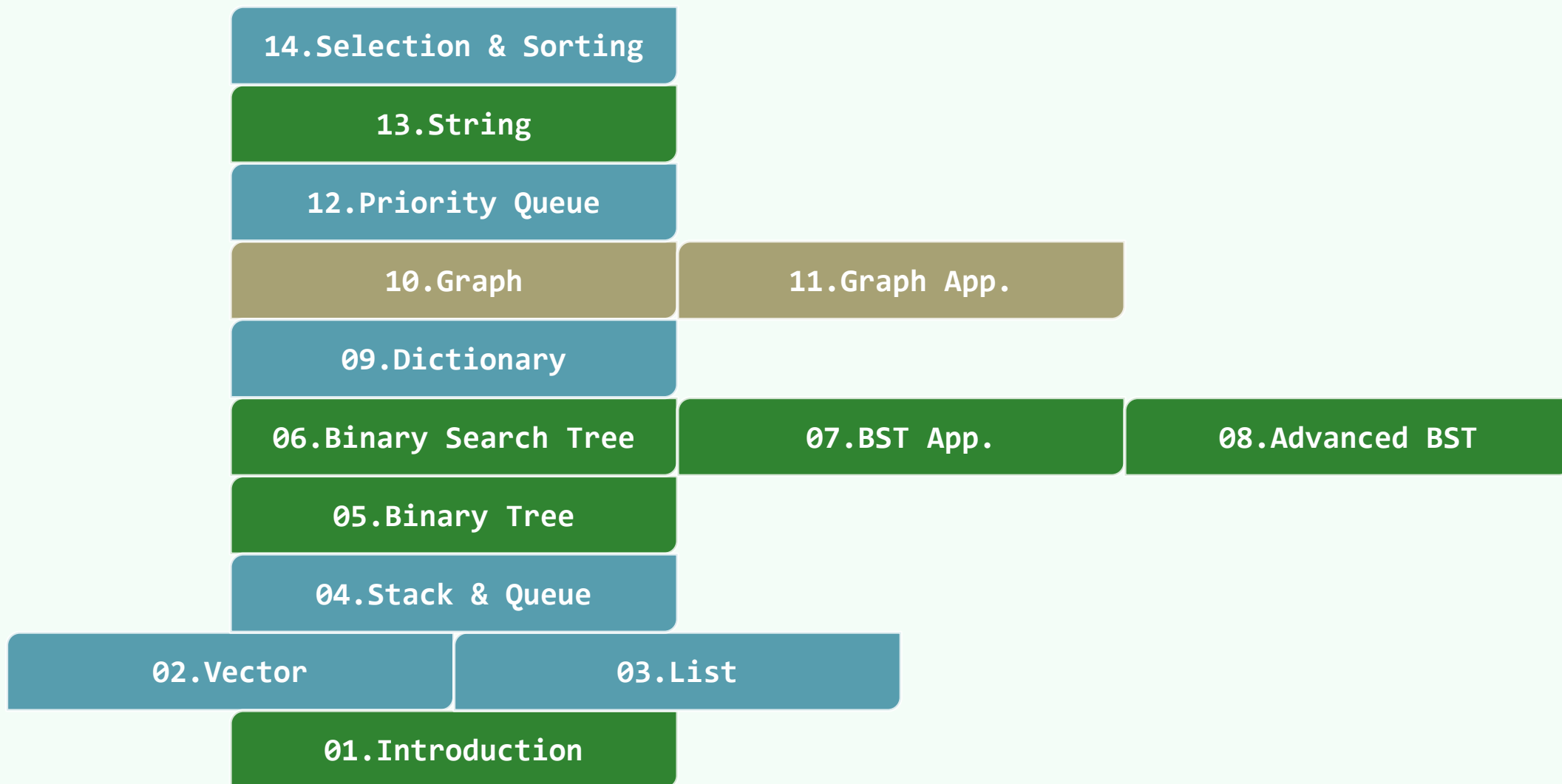
- **C/C++语言**： 对象/属性/方法、类/构造/析构、继承/派生、重载、重写、虚方法、模板
- **离散数学**： 集合、偏序、良序、数学归纳法、
级数、递归/递推、排列/组合、Stirling逼近、Fib数、Catalan数
- **概率与统计**： 概率、随机分布、数学期望、期望值的线性律、几何分布、二项式分布
- (涉及数学之处，我们会尽可能使用**初等**的方法)

❖ **志趣驱动**： 有志于用计算机创新技术、服务人类；对算法有强烈的好奇心；勤于钻研，乐于探究

❖ **平和心态**： 体验学习的**过程**，看重实在的**收获**

❖ **科学方法**： **理解、记忆**不要偏废；**思考、交流**双管齐下；**具体、抽象**迭代深化；**继承、综合**循序渐进

课程知识体系



实践训练：既入宝山，焉能空返

- ❖ **PA**：针对实际问题，综合运用讲授内容
- ❖ **LAB**：全面了解测试与调试过程，掌握相关方法与技巧
深入洞悉各类数据结构特点，学会灵活加一运用
- ❖ 尽早阅读**帮助文档**
了解实验**形式**、相关**纪律**及基本**技术**



<https://dsa.cs.tsinghua.edu.cn/oj>

A) 用email注册账号

Tsinghua Online Judge

Department of Computer Science and Technology, Tsinghua University

Welcome!

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

B) 从网络学堂获取邀请码加入课堂

Join Private Course

Code:

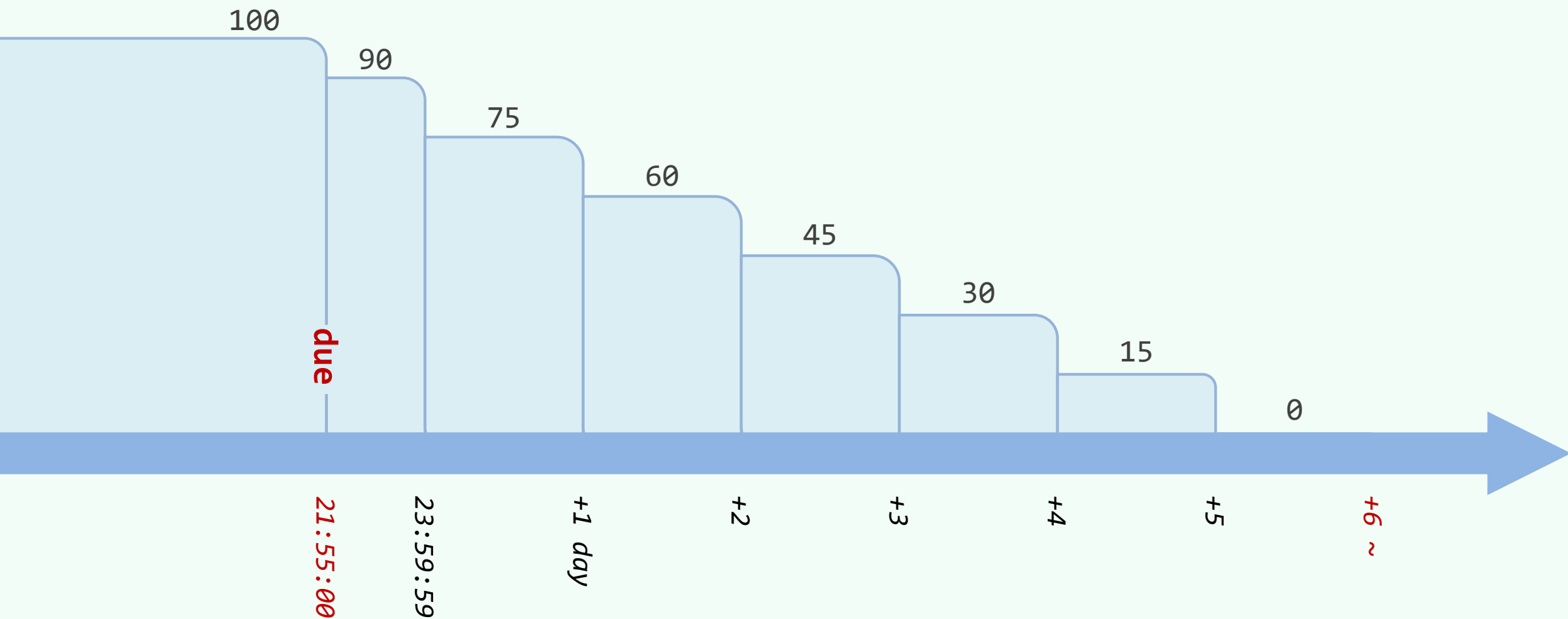
纪律与规范

- ✓ 须独立完成，否则后果严重
 - **双方同论**不作区分，查重对象覆盖**历届**的选课学生
 - 以**相似程度**作为判定依据；查重范围覆盖**代码与文档**
- ✓ 流程 = 查重**系统**初筛 + **人工**核验与抽查 + 查看**HonorCode** + 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 ...
 - 复议时只需补充可能与HonorCode效力**等同**或**接近**的信息，**其余内容均不予**采纳
- ✓ **不可以**交流且**有责任**保管好：**源代码**；可导致相似的**伪代码**；实验报告等参与考核的**文档**
- ✓ **可以**交流：题意理解、解题思路、算法与数据结构的设计与选用方案、测试用例
- ✓ 尺度拿捏不准？涉及的人员、文献、资源等，尽可能在**HonorCode**中列出吧
 - 只要注明得**客观、准确、完整**，就至少不会倒扣

如何复议

- ❖ **其实无效：**
 - x “我是独立完成的，OJ上有历史记录的；**是ta**抄袭了我的代码”
 - x “我把代码给ta看时，ta**承诺过**只为启发思路”
 - x “我肯定**没有**给任何人看过我的代码”
 - x “我和ta完全**没有**交集，怎么可能会代码相似”
 - x “我**已经**在HonorCode中列举过参考源的，为什么还要被罚”
 - x “ta承认偷拍、窃取了代码，我毫**不知情**”
 - x “我们自己用查重工具，检测出来的相似度并**不高**呀”
 - x ...
- ❖ **或许有效：**
 - o “我在实验报告、代码注释中，曾做过**类似于**HonorCode的交待”

作业补交



辅导：形式与原则

习题课

- 带有语音的讲稿，随进度通过网络学堂陆续发布
- 讲解**编程作业**的思路及要点，测试、调试技巧

日常交流

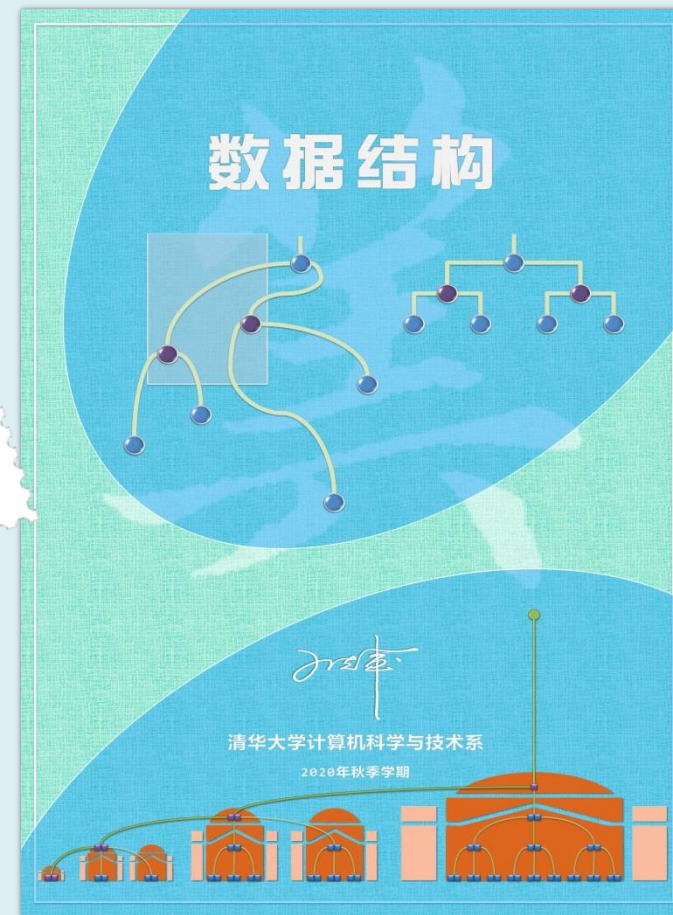
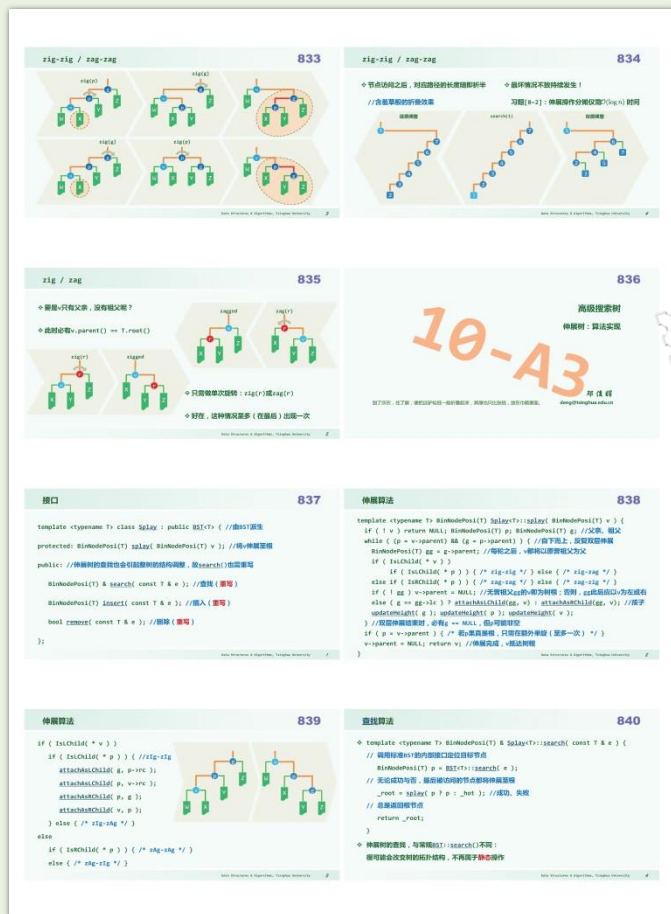
- 主要依托网络学堂值班助教会**及时响应**
- 共性问题上**讨论区**
- 个性问题上**答疑区**

个别答疑

- 线上或线下形式，不定期
- 助教：针对你在完成**编程作业**时遇到的技术性问题
- 老师：针对**课上**讲授内容

- 提问之前，应在课上**听过**对应知识点的讲授，已**测试过**示例代码包中对应的算法，**核对过**教材勘误表
- 提问之前，需已掌握课堂讲授的**基本**内容、做过一定的独立**思考**和**实践**尝试
- 提问时，请尽可能**简洁、清晰、规范**地描述问题

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

慕课：学堂在线或其它视频平台

- ✓ **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

教材: <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；
年头太**久远**，太多处已做**更新**
校内学生**强烈不建议**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**答疑区**反馈

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, 起始于顶点的DFS经
- ☐
☐ 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
- ☐

- ☐ 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
- ☐
☐ 在插入操作后若红黑树需高度增加,
- ☐
☐ 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
- ☐
☐ 无论是单独借助BCI表或OSI表, 均
- ☐
☐ 对规模为n的AVL树做一次插入操
- ☐

数据结构 (30240184 · 2018秋)

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

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, 文本串的

数据结构 (30240184 · 2018秋)

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
 BinNodePosi<T> BinNodePosi<T> prePos, T inPos, int n) {
 if (n < 1) return NULL;
 k = search(pre[0], in, n); // locate
 return new BinNode(pre[a], //key
 reconstruct(prePos, inPos, k-1),
 reconstruct(prePos+k, inPos, n-k));
}

A) 试补全递归调用处缺失的参数; B) 设每次递归

9. Bitmap & QuadTree

各像素或黑或白的图像, 也称作位图; 正方

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

Data Structures (30240184, Fall 2020)

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的

试证明: 若二叉树的先序遍历序列为 $\{v_0,$

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

数据结构 (30240184 · 2014秋)

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的查找。若在某系统中经测量确认, 三种算法的时间复杂度常系数约为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)。

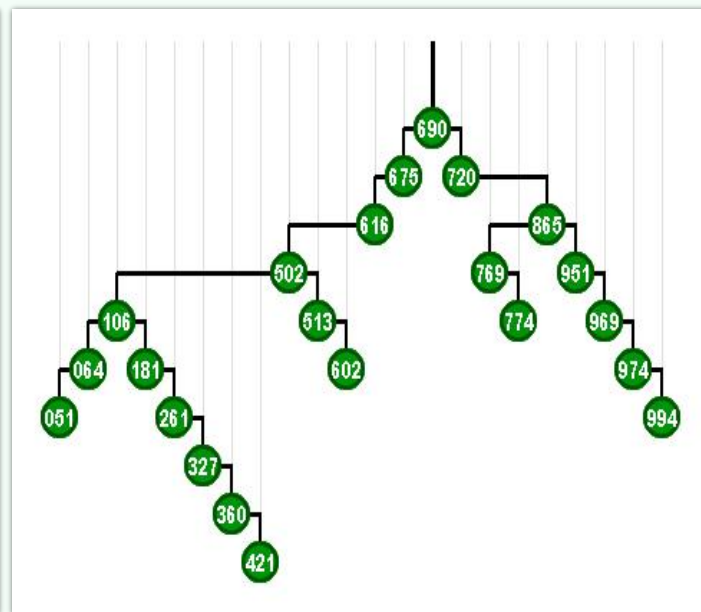
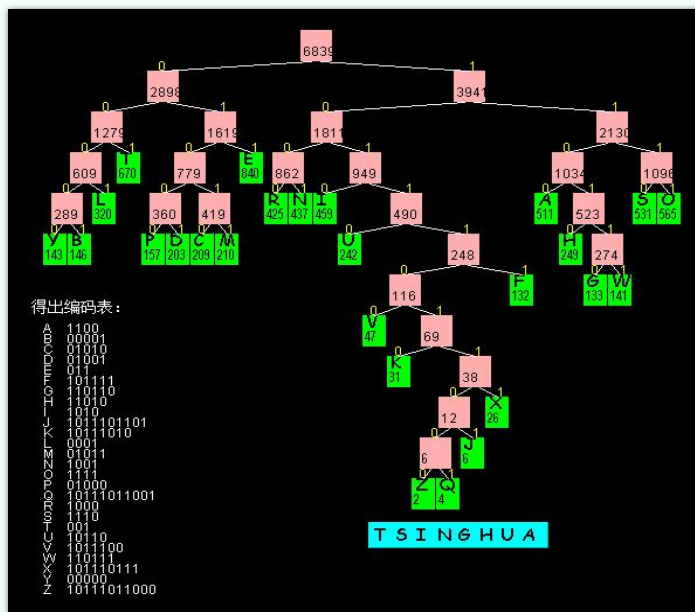
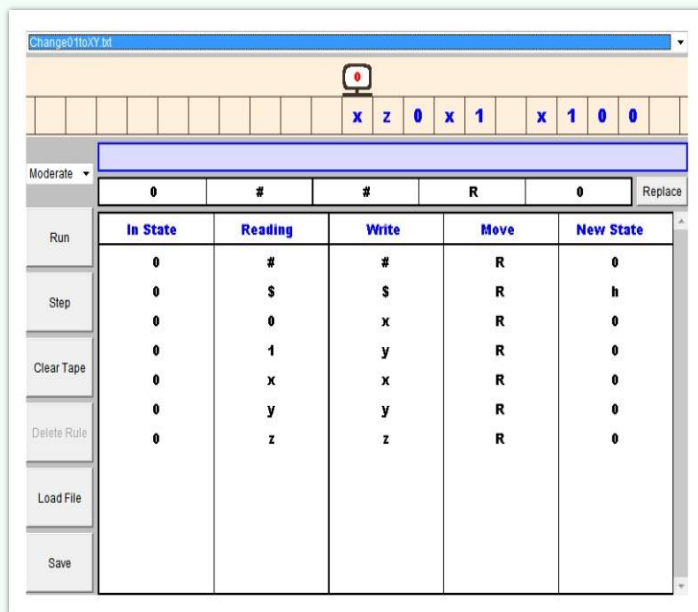
Data Structures (30240184, Fall 2019)

3 / 4

Midterm 11/07/2019

教学演示 (Applet版)

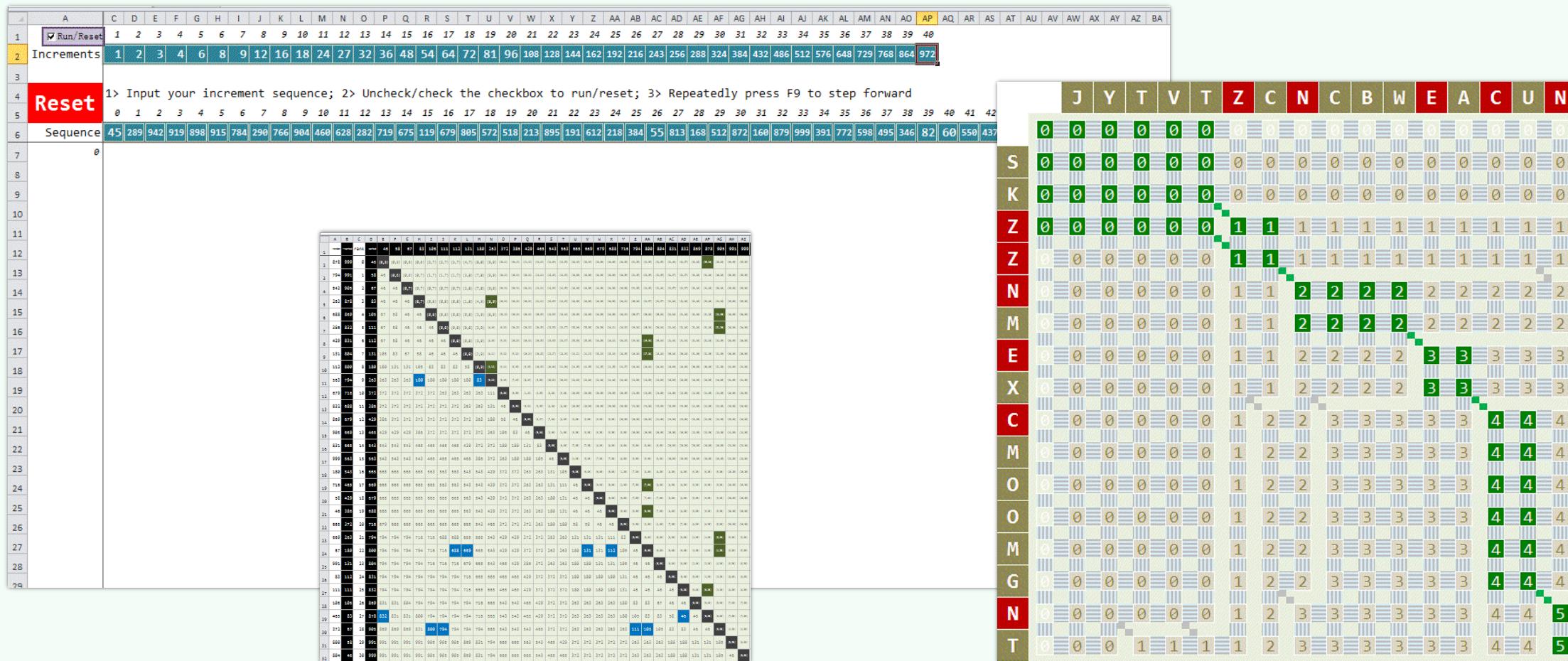
- ✓ 可借助AppletViewer.exe播放 (建议将index.applet与之绑定)



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

教学演示 (Excel版)

✓ 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（与其它环境**未必**兼容）
- ✓ 可直接编译执行，多通过**命令行**设置输入
- ✓ 代码风格针对**教学**需求，**请勿**简单模仿
- ✓ **声明**后可用于PA，对可能含有的**BUG**自己负责

```
Searching for 0277 ... Not found
Removing 0381 ... Entry not exists
Inserting < 0212, Q> ... Done
Searching for 0144 ... Found with F
struct Skiplist<int,char>[13631200]*7*6:
Qlist[9024]*000: -----
Qlist[9336]*001: -----< 0090: D>-----
Qlist[9240]*002: -----< 0090: D>-----< 0284: M>--
Qlist[9192]*003: -----< 0090: D>--< 0144: F>-----< 0284: M>--
Qlist[8976]*004: -< 0064: B>--< 0090: D>--< 0144: F>-----< 0284: M>--
Qlist[8976]*005: -< 0064: B>--< 0090: D>--< 0144: F>-----< 0268: K>--< 0284: M>--
Qlist[8976]*006: -< 0064: B>--< 0090: D>--< 0144: F>--< 0212: Q>--< 0268: K>--< 0284: M>--
Searching for 0212 ... Found with Q
Searching for 0123 ... Not found
```

```
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
```

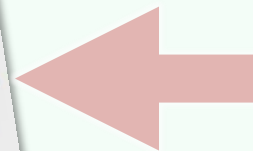
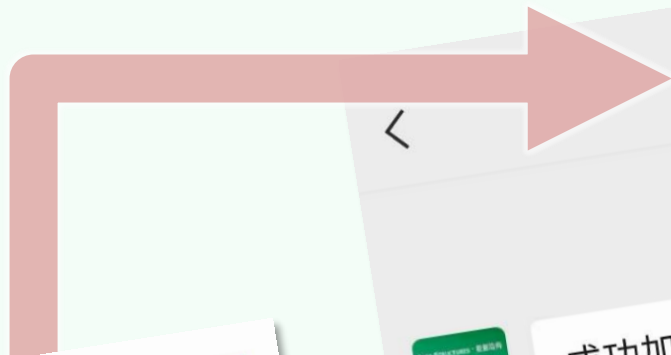
```
Inserting 2307 ... Done
class BTree<int>[4128208]*4:
3644 *>└ 3644
2307 *>└ 2307
1783 *>└ 1783
0178 *>└ 0178
```

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

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


A

扫码关注



B

加入频道