1. Reference: 王珊老师
2. Web course: <https://www.bilibili.com/video/av6239731?from=search&seid=15262918976300524527>
3. // video lecture;
4. Ep 01: 概述1; 教材推荐; (2019-1-10)
5. Ep 02: 数据库系统概述; 数据库的地位; 数据, 描述事物的符号记录; 数据库, 存放数据的地方; 完整性, 正确性, 并发性; DBA职业; 数据独立性; 物理独立性; 逻辑独立性; (2019-1-11)
6. Ep 03: 数据模型; 数据结构; 数据的完整性约束条件; 工厂物质管理E-R模型; 非关系模型(格式化模型); 层次数据模型的数据结构; 教员学生数据库模型; 层次模型的完整性约束; 子女-兄弟链接法示例; 层次模型的优缺点; 网状模型; 网状数据模型的完整性约束; 网状数据模型的存储结构示例; 网状模型的优缺点; 典型网状数据库系统; 关系模型; 关系数据模型的数据结构; 关系模型的数据操纵; 典型的关系数据库系统; (2019-1-11)
7. Ep 04: 数据库系统模式的概念; 模式的一个实例; 学校教学管理数据库的模式; 型与值的概念; 外模式; 数据库的二级映像功能; 逻辑独立性; 模式/内模式映像; (2019-1-27)
8. Ep 05: 数据库系统的组成; 对硬件的要求; 软件; 人员; 各种人员的数据视图; 数据库管理员DBA; 系统分析员和数据库设计人员; 应用程序员; 用户; 单用户数据库系统; 主从式结构的数据库系统; 分布式数据库系统; 客户/服务器结构的数据库系统; 并行数据库系统; (2019-1-27)
9. Ep 06: 关系; 域; 笛卡尔积; 元祖; 基数; 主码; 主属性; 基本关系; 查询表; 视图表; 基本关系的性质; 关系模式; 定义关系模式; 关系数据库; 关系数据库模式; (2019-1-27)
10. Ep 07: 关系操作; 基本关系操作; 关系数据库语言的分类; (2019-1-27)
11. Ep 08: 关系的完整性; 实体完整性; 不能取空值; 实体是可区分的; 参照完整性; 关系间的引用; 外码; (2019-1-27)
12. Ep 09: 关系代数; 用户定义的完整性; 并操作; 差; 交; 笛卡尔积; 专门的关系运算; 选择selection; 投影projection; 连接join; 除division; 综合举例; (2019-1-28)
13. Ep 10: 关系演算; 检索操作; 简单检索; 限定的检索; 带排序的检索; 指定返回元祖的条数的检索; 用元组变量的检索; 用存在量词的检索; 带有多个关系的表达式的检索; (2019-1-29)
14. ep 11: SQL概述; SQL的产生和发展; SQL的特点; 高度非过程化; 面向集合的操作方式; 以同一种语法提供多种使用方式; 语言简单, 易学易用; SQL的基本概念; 基本表; 存储文件; 视图表; (2019-1-29)
15. Ep 12: 数据定义; 创建create; schema和view没有alter; 定义模式; 基本表的定义删除修改; 数据字典; 模式与表; 修改基本表; 删除基本表; 索引的建立和删除; 建立索引; (2019-1-31)
16. Ep 13: 数据查询; select from where; 比较大小; 确定范围; 确定集合; 字符匹配; 涉及空值的查询; 多重条件查询; 单表查询; ORDER BY; 聚集函数; GROUP BY子句; (2018-4-8)
17. Ep 14: 数据查询(2); 连接查询; 等值与非等值连接查询; 自身连接; 外连接; 多表连接; 嵌套查询; 不相关子查询; 带有IN谓词的子查询; 带有比较运算符的子查询; 带有ANY或ALL谓词的子查询; 带有EXISTS谓词的子查询; 集合查询; 关于GROUP BY子句; 关于Having的条件子句; 关于ORDER BY子句; 集合查询; 一个查询的多种表达; 子查询用连接查询表达; select语句的一般形式; (2019-2-1)
18. ep 15: 数据更新; 插入数据; 插入元组; 插入子查询结果; 修改数据; 修改某一个元组的值; 修改多个元组的值; 带子查询的修改语句; 删除数据; (2019-2-1)
19. ep 16: 视图; 定义视图; 建立视图; 删除视图; 查询视图; 更新视图; 视图的作用; (2019-2-1)
20. Ep 17: example 3.2, 给出证明栈操作是否合法的准则; 第一个子列s>=x, 整个序列s=x; example 3.6, 用while和stack代替递归; 识别一个读入的反对称序列, 用stack; example 3.15, 判断是否回文, 比较栈顶和队头; example 4.7, 从串S中删除所有和串T相同的子串; example 4.11, 求最长重复子串; (2019-2-2)
21. Ep 18: 数据库安全性概述; 数据库安全性; (2019-2-2)
22. ep 19: 数据库安全性控制; 数据库安全性控制概述; 用户身份鉴别; 存取控制; 自主存取控制方法; 授权: 授予和回收; 数据库角色; 强制存取控制方法; (2019-2-4)
23. Ep 20: 视图机制; (2019-2-4)
24. Ep 21: 审计, 数据加密以及统计数据安全性; 数据加密; (2019-2-4)
25. Ep 22: 实体完整性; 数据的正确性; 数据的完整性; 数据的安全性; 提供定义完整性约束条件的机制; 提供完整性检查的方法; 违约处理; (2019-2-4)
26. Ep 23: 参照完整性; 参照完整性定义; 参照完整性检查和违约处理; (2019-2-4)
27. ep 24: 用户定义的完整性; 属性上的约束条件; 元组上的约束条件; (2019-2-4)
28. Ep 25: 完整性约束命名子句; 完整性约束命名子句; (2019-2-6)
29. Ep 26: 域中完整性限制; 断言assertion; (2019-2-6)
30. ep 27: 触发器; 定义触发器; 激活触发器; 删除触发器; (2019-2-6)
31. ep 28: 问题的提出; key的形式化定义; 关系数据理论; 数据依赖; 函数依赖; 传递函数依赖; 码; (2019-2-6)
32. Ep 29: 规范化; 第一范式; 第二范式; 第三范式; BC范式; 多值依赖和第四范式; 多值依赖; 多值依赖的对称性; (2019-2-7)
33. ep 30: 数据依赖的公理系统; 基本思路; 函数依赖集等价; 最小依赖法; (2019-2-7)
34. Ep 31: 模式的分解; (2019-2-7)
35. Ep 32: 数据库设计概述; 数据库设计的特点; 数据库设计方法; 数据库设计的基本步骤; 数据库设计中的各级模式; (2018-5-1)
36. Ep 33: 需求分析; 需求分析的任务; 需求分析的方法; 数据字典; (2019-2-7)
37. Ep 34: 概念结构设计; 属性冲突; 消除冗余的方法, 分析方法; (2019-2-7)
38. Ep 35: 逻辑结构设计; E-R图向关系模型的转换; 数据模型的优化; 设计用户字模式; (2019-2-8)
39. Ep 36: 数据库的物理设计; 数据库物理设计的内容和方法; 关系模式存取方法选择; 确定数据库的存储结构; 评价物理结构; (2019-2-8)
40. Ep 37: 嵌入式SQL; (2019-2-11)
41. Ep 38: 存储过程; (2019-2-11)
42. Ep 39: 关系数据库系统的查询处理; 查询分析; 查询检查; 查询优化; 查询执行; 关系数据库的查询优化; 选择操作的实现; 连接操作的实现; (2019-2-11)
43. Ep 40: 关系数据库系统的查询优化; (2019-2-11)
44. Ep 41: 代数优化; 等价交换规则; 交换律; 结合律; (2019-2-11)
45. Ep 42: 物理优化; 基于启发式规则的存取路径选择优化; 选择操作的启发式规则; 连接操作的启发式规则; 基于代价的优化; 数据库恢复技术; (2019-2-17)
46. Ep 43: 事务和数据库恢复概述; 事务; 原子性; 一致性; 隔离性; 持续性; (2019-2-17)
47. Ep 44: 故障种类和恢复实现技术; 事物内部的故障; 系统故障; 使用日志undo; 介质故障; 计算机病毒; (2019-2-18)
48. Ep 45: 恢复策略; 数据转储; 登记日志文件; (2019-2-18)
49. Ep 46: 具有检查点的恢复技术和数据库镜像; 问题的提出; 检查点技术; 利用检查点的恢复策略; 数据库镜像; (2019-2-18)
50. Ep 47: 并发控制概述; 丢失修改; 不可重复读; 读脏数据; 并发控制的主要数据; 封锁; 时间戳; 乐观控制法; 多版本并发控制; (2019-2-18)
51. Ep 48: 封锁; 排他锁; 共享锁; (2019-2-18)
52. Ep 49: 活锁和死锁; 活锁; 死锁; 死锁的预防; 死锁的诊断与解除; (2019-2-18)
53. Ep 50: 并发调度的可串行性; (2019-2-18)
54. Ep 51: 并发控制(3); (2019-2-18)
55. // text book;
56. Chapter 1: 绪论;
57. Node 1.1: 数据库系统概述;
58. List 1.1.1: 数据库的4个基本概念;
59. 数据data; 数据库data base; 数据库管理系统DBMS; 数据库系统DBS; (2019-1-14)
60. List 1.1.2: 数据管理技术的产生和发展;
61. 人工管理阶段; 文件系统阶段; 数据库系统阶段; (2019-1-14)
62. List 1.1.3: 数据库系统的特点;
63. 数据结构化; 数据共享性高, 冗余度低且易扩充; 数据独立性高; (2019-1-14)
64. Node 1.2: 数据模型;
65. 数据模型, 对现实世界数据特征的抽象; 数据模型是数据库的核心和基础; (2019-1-14)
66. List 1.2.1: 两类数据模型;
67. 概念模型; 逻辑模型和物理模型; 概念模型; (2019-1-16)
68. List 1.2.2: 概念模型;
69. 实体, 客观存在并可相互区别的事物; 属性, 实体所具有的某一特性; 码key, 唯一标识实体的属性集; 实体型entity type; 实体集; 联系relationship; 实体联系方法; (2019-1-16)
70. List 1.2.3: 数据模型的组成要素;
71. 数据结构; 数据操作; 数据的完整性约束条件; (2019-1-16)
72. List 1.2.4: 常用的数据模型;
73. List 1.2.5: 层次模型;
74. 层次模型的数据结构; 层次模型的数据操纵和完整性约束; 层次模型的优缺点; (2019-1-16)
75. List 1.2.6: 网状模型;
76. 网状模型的数据结构; 网状模型的数据操纵和完整性约束; 网状模型的优缺点; (2019-1-17)
77. List 1.2.7: 关系模型;
78. 关系模型的数据结构; 关系模型的数据操纵和完整性约束; 关系模型的优缺点; (2019-1-17)
79. Node 1.3: 数据库系统的结构;
80. List 1.3.1: 数据库系统模式的概念;
81. 模式schema; 实例instance; (2019-1-17)
82. List 1.3.2: 数据库系统的三级模式结构;
83. 模式schema; 外模式external schema; 内模式internal schema; (2019-1-17)
84. List 1.3.3: 数据库的二级映像功能与数据独立性;
85. 外模式/模式映像; 模式/内模式印象; (2019-1-17)
86. Node 1.4: 数据库系统的组成;
87. 硬件平台及数据库; 软件; 人员; (2019-1-18)
88. Chapter 2: 关系数据库;
89. Node 2.1: 关系数据结构及形式化定义;
90. List 2.1.1: 关系;
91. 域domain; 笛卡尔积cartesian product; 关系relation; 候选码candidate key; 基本关系; 查询表; 视图表; (2019-1-18)
92. List 2.1.2: 关系模式;
93. List 2.1.3: 关系数据库;
94. List 2.1.4: 关系模型的存储结构;
95. Node 2.2: 关系操作;
96. List 2.2.1: 基本的关系操作;
97. List 2.2.2: 关系数据语言的分类;
98. 关系代数; 关系演算; 结构化查询语言SQL; (2019-1-18)
99. Node 2.3: 关系的完整性;
100. 实体完整性; 参照完整性; 用户定义的完整性; (2019-1-18)
101. List 2.3.1: 实体完整性;
102. 主属性不能取空; (2019-1-18)
103. List 2.3.2: 参照完整性;
104. 外码foreign key; 参照完整性规则; (2019-1-18)
105. List 2.3.3: 用户定义的完整性;
106. Node 2.4: 关系代数;
107. 对关系的运算来表达查询; (2019-1-18)
108. List 2.4.1: 传统的集合运算;
109. 并union; 差except; 交intersection; 笛卡儿积cartesian product; (2019-1-18)
110. List 2.4.2: 专门的关系运算;
111. 选择selection; 投影projection; 连接join; 除运算divide; 关系代数表达式; (2019-1-18)
112. Node 2.5: 关系推算;
113. List 2.5.1: 元组关系演算语言ALPHA;
114. 检索操作; 更新操作; 修改操作; 插入操作; 删除; (2019-1-19)
115. List 2.5.2: 元组关系演算;
116. 元组演算表达式; (2019-1-19)
117. List 2.5.3: 域关系语言QBE;
118. 检索操作; 更新操作; 修改操作; 插入操作; 删除操作; (2019-1-19)
119. Chapter 3: 关系数据库标准语言SQL;
120. Node 3.1: SQL概述;
121. List 3.1.1: SQL的产生与发展;
122. List 3.1.2: SQL的特点;
123. 综合统一; 高度非过程化; 面向集合的操作方式; 以同一种语法结构提供多种使用方式; 语言简洁, 易学易用; (2019-1-19)
124. List 3.1.3: SQL的基本概念;
125. Node 3.2: 学生-课程数据库;
126. Node 3.3: 数据定义;
127. List 3.3.1: 模式的定义与删除;
128. 定义模式; 删除模式; (2019-1-19)
129. List 3.3.2: 基本表的定义/删除/修改;
130. 定义基本表; 数据类型; 模式与表; 修改基本表; 删除基本表; (2019-1-19)
131. List 3.3.3: 索引的建立与删除;
132. 建立索引; 修改索引; 删除索引; (2019-1-19)
133. List 3.3.4: 数据字典;
134. Node 3.4: 数据查询;
135. List 3.4.1: 单表查询;
136. 选择表中的若干列; 选择表中的若干元组; ORDER BY语句; 聚集函数; GROUP BY子句; (2019-1-20)
137. List 3.4.2: 连接查询;
138. 等值与非等值连接查询; 自身连接; 外连接; 多表连接; (2019-1-20)
139. List 3.4.3: 嵌套查询;
140. 带有IN的谓词的子查询; 不相关子查询; 相关子查询; 带有比较运算符的子查询; 带有ANY(SOME)或ALL谓词的子查询; 带有EXISTS谓词的子查询; (2019-1-20)
141. List 3.4.4: 集合查询;
142. List 3.4.5: 基于派生表的查询;
143. List 3.4.6: SELECT语句的一般格式;
144. 目标表达式的可选格式; 聚集函数的一般格式; WHERE子句的条件表达式的可选格式; (2019-1-20)
145. Node 3.5: 数据更新;
146. List 3.5.1: 插入数据;
147. 插入元组; 插入子查询的结果; (2019-1-20)
148. List 3.5.2: 修改数据;
149. 修改某一个元组的值; 修改多个元组的值; 带子查询的修改语句; (2019-1-20)
150. List 3.5.3: 删除数据;
151. 删除某一个元组的值; 删除多个元组的值; 带子查询的删除语句; (2019-1-20)
152. Node 3.6: 空值的处理;
153. 空值的产生; 空值的判断; 空值的约束条件; 空值的算术运算比较运算逻辑运算; (2019-1-20)
154. Node 3.7: 视图;
155. List 3.7.1: 定义视图;
156. 建立视图; 行列子集视图; 分组视图; 删除视图; (2019-1-20)
157. List 3.7.2: 查询视图;
158. 视图消解; (2019-1-20)
159. List 3.7.3: 更新视图;
160. List 3.7.4: 视图的作用;
161. 视图能够简化用户的操作; 视图使用户能以多种角度看待同一数据; 视图对重构数据库提供了一定程度的逻辑独立性; 视图能够对机密数据提供安全保护; 适当利用视图可以更清晰地表达查询; (2019-1-20)
162. Chapter 4: 数据库安全性;
163. Node 4.1: 数据库安全性概述;
164. List 4.1.1: 数据库的不安全因素;
165. 非授权用户对数据库的恶意存取和破坏; 数据库中重要或敏感的数据被泄漏; 安全环境的脆弱性; (2019-1-20)
166. List 4.1.2: 安全标准简介;
167. Node 4.2: 数据库安全性控制;
168. List 4.2.1: 用户身份鉴别;
169. 静态口令鉴别; 动态口令鉴别; 生物特征鉴别; 智能卡鉴别; (2019-1-21)
170. List 4.2.2: 存取控制;
171. 定义用户权限; 合法权限检查; (2019-1-21)
172. List 4.2.3: 自主存取控制方法;
173. List 4.2.4: 授权: 授予与收回;
174. GRANT; REVOKE; 创建数据库模式的权限; (2019-1-21)
175. List 4.2.5: 数据库角色;
176. 角色的创建; 给角色授权; 将一个角色授予其他的角色或用户; 角色权限的收回; (2019-1-21)
177. List 4.2.6: 强制存取控制方法;
178. Node 4.3: 视图机制;
179. Node 4.4: 审计;
180. 审计事件; 审计功能; AUDIT语句和NOAUDIT语句; (2019-1-22)
181. Node 4.5: 数据加密;
182. 存储加密; 传输加密; (2019-1-22)
183. Node 4.6: 其他安全性保护;
184. 推理控制; 隐蔽信道; (2019-1-22)
185. Chapter 5: 数据库完整性;
186. 提供定义完整性约束条件的机制; 提供完整性检查的方法; 进行违约处理; (2019-1-22)
187. Node 5.1: 实体完整性;
188. List 5.1.1: 定义实体完整性;
189. List 5.1.2: 实体完整性检查和违约处理;
190. Node 5.2: 参照完整性;
191. List 5.2.1: 定义参照完整性;
192. List 5.2.2: 参照完整性检查和违约处理;
193. 拒绝执行; 级联操作; 设置为空值; (2019-1-22)
194. Node 5.3: 用户定义的完整性;
195. List 5.3.1: 属性上的约束条件;
196. 属性上约束条件的定义; 不允许空值; 列值唯一; 用check短语指定列值; 属性上约束条件的检查和违约处理; (2019-1-23)
197. List 5.3.2: 元组上的约束条件;
198. 元组上的约束条件的定义; 元组上约束条件的检查和违约处理;
199. Node 5.4: 完整性约束命名子句;
200. 完整性约束命名子句; 修改表中的完整性限制; (2019-1-23)
201. Node 5.5: 域中的完整性限制;
202. Node 5.6: 断言;
203. 创建断言的语句格式; 删除断言的语句格式;
204. Node 5.7: 触发器;
205. List 5.7.1: 定义触发器;
206. List 5.7.2: 激活触发器;
207. List 5.7.3: 删除触发器;
208. 第二篇: 设计与应用开发篇;
209. Chapter 6: 关系数据理论;
210. Node 6.1: 问题的提出;
211. 数据依赖; 数据冗余; 更新异常; 插入异常; 删除异常; (2019-1-23)
212. Node 6.2: 规范化;
213. List 6.2.1: 函数依赖;
214. 完全函数依赖, 部分函数依赖; 传递函数; (2019-1-24)
215. List 6.2.2: 码;
216. 候选码; (2019-1-24)
217. List 6.2.3: 范式;
218. 规范化; (2019-1-24)
219. List 6.2.4: 2NF;
220. 插入异常; 删除异常; 修改复杂; (2019-2-18)
221. List 6.2.5: 3NF;
222. List 6.2.6: BCNF;
223. ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
224. Chapter 6: 树和二叉树;
225. Ch 6.1: 子树; 结点, degree; leaf, child, parent, sibling; 祖先, 子孙; 层次, 堂兄弟, 深度; 有序树, 无序树; 森林; (2018-5-12)
226. Ch 6.2: 二叉树; 二叉树的性质; 二叉树的顺序存储; 二叉树的链式存储; (2018-5-13)
227. Ch 6.3: 遍历二叉树; Preorder, Inorder, Postorder; 前缀表示, 中缀表示, 后缀表示; 中序遍历的非递归算法; 按先序序列建立二叉链表; 遍历二叉树时间复杂度O(n), 空间复杂度O(n); 线索链表, 线索, 线索二叉树, 线索化; (2018-5-13)
228. Ch 6.4: 树的存储结构, 双亲表示法, 孩子表示法, 孩子兄弟表示法; 树和二叉树的对应关系; 先根遍历树, 后根遍历树; 先序遍历森林, 中序遍历森林; (2018-5-14)
229. Ch 6.5: 等价关系, 等价类; example 6.1, 求等价类; (2018-5-14)
230. Ch 6.6: 路径长度, 带权路径长度; 最优二叉树, 赫夫曼树; 构造赫夫曼树, 赫夫曼算法; 赫夫曼编码; 出现次数较多的字符采用尽可能短的编码; example 6.2, 设计赫夫曼编码; (2018-5-15)
231. Ch 6.7: 回溯法和树的遍历; example 6.3, 求幂集; example 6.4, 求4皇后问题的所有布局; (2018-5-15)
232. Ch 6.8: 树的计数; example 6.5, 从前序序列和中序序列构造二叉树; (2018-5-15)
233. Chapter 7;
234. Ch 7.1: 顶点, 弧尾, 弧头, 有向图, 边, 无向图; 完全图, 有向完全图; 稀疏图, 稠密图; 权, 网; 邻接点, 依附, 关联; 度, 入度, 出度; 路径, 回路或环, 简单路径; 连通, 连通图, 连通分量; 强连通图, 强连通分量; 生成树; 有向树, 生成森林; (2018-5-12)
235. Ch 7.2: 邻接矩阵; 利用邻接矩阵求顶点的度; 网的邻接矩阵; 邻接表, 每个顶点与之相邻的顶点; 逆邻接表; (2018-5-12)
236. Ch 7.3: 图的遍历; 深度优先搜索, 类似于先根遍历; 广度优先搜索, 类似于按层次遍历; (2018-5-12)
237. Ch 7.4: 图的连通性; 非连通图, 需从多个顶点出发进行搜索; 极小连通子图, 深度优先生成树, 广度优先生成树; 非连通图的生成森林; 深度优先生成森林; 利用十字链表, 求有向图的强连通分量; 最小生成树; 最小生成树的MST性质; Prim算法; Kruskal算法; 关节点(割点), 重连通图, 连通度; (2018-5-12)
238. Ch 7.5: 有向无环图; 拓扑排序; 选一个没有前驱的结点并输出, 从图中删除该顶点和所有以它为尾的弧; 顶点表示活动的网(AOV网); 边表示活动的网(AOE网); 关键路径; (2018-5-12)
239. Ch 7.6: 最短路径; Dijkstra算法; D[i], 当前所找到的从始点v到终点vi的最短路径长度; Floyd算法; (2018-5-12)
240. Chapter 9;
241. 查找表, 静态查找表, 动态查找表; 关键字, 主关键字, 次关键字; 查找, 查找成功, 查找不成功; (2018-5-18)
242. Ch 9.1: 静态查找表; 顺序查找; 平均查找长度; 有序表的查找; 折半查找; 折半查找的性能分析; 静态树表的查找; ;;;;;
243. Ch 9.3: 哈希函数; 冲突, 同义词; 哈希表, 散列, 哈希地址或散列地址; 哈希函数的构造方法; 均匀的哈希函数; 直接定址法; 数字分析法; 平方取中法; 折叠法; 除留余数法; 随机数法; 处理冲突的方法; 开放定址法; 再哈希法; 链地址法, example 9-3, 链地址法构造的哈希表; 建立一个公共溢出区; 哈希表的查找及其分析; 哈希表的装填因子; (2018-5-19)
244. Chapter 10;
245. Ch 10.1: 排序方法是稳定的, 排序方法是不稳定的; 内部排序, 外部排序; 简单的排序, 先进的排序, 基数排序; 比较两个关键字的大小, 将记录葱一个位置移动到另一个位置; 线性表, 链表排序, 地址排序; (2018-5-21)
246. Ch 10.2: 直接插入排序; ;;;;;;;;;;;;;;;;;;
247. 王道论坛考研数据结构;
248. Chapter 1: 绪论;
249. 分析算法的时间复杂度和空间复杂度; (2018-9-11)
250. Node 1.1: 数据结构的基本概念;
251. List 1.1.1: 基本概念和术语; 数据; 一个数据元素可由若干个数据项组成; 数据对象, 具有相同性质的数据元素的集合; 数据类型, 原子类型, 结构类型, 抽象数据类型; 抽象数据类型, 数据对象, 数据关系, 基本操作集; 数据结构, 逻辑结构, 存储结构, 数据的运算; (2018-9-11)
252. List 1.1.2: 数据结构的三要素; 数据的逻辑结构; 线性结构, 线性表; 非线性结构, 集合, 树, 图; 数据的存储结构; 顺序存储; 链式存储; 索引存储; 散列存储; 数据的运算; (2018-9-11)
253. List 1.1.3: 选择4, 循环队列是顺序表表示的; 选择7, 结点内一定连续;
254. Node 1.2: 算法和算法评价;
255. List 1.2.1: 算法的基本概念; 有穷性, 确定性, 可行性, 输入, 输出; 正确性, 可读性, 健壮性, 效率与低存储量需求; (2018-9-11)
256. List 1.2.2: 算法效率的度量; 时间复杂度; 最坏时间复杂度, 平均时间复杂度, 最好时间复杂度; 空间复杂度; (2018-9-11)
257. List 1.2.3: 选择6; 选择12;
258. Chapter 2: 线性表;
259. Node 2.1: 线性表的定义和基本操作;
260. List 2.1.1: 线性表是具有相同数据类型的数据元素的有限序列; 每个元素有且仅有一个前驱, 一个后继; (2018-9-11)
261. List 2.1.2: 线性表的基本操作; (2018-9-11)
262. Node 2.2: 线性表的顺序表示;
263. List 2.2.1: 顺序表的定义, 线性表的顺序存储; 一维数组可以是静态分配的, 也可以是动态分配的; (2018-9-11)
264. List 2.2.2: 顺序表上基本操作的实现; 插入操作; 删除操作; 按值查找; 顺序表存储空间=表长\*sizeof(元素类型); (2018-9-15)
265. page 18, question 12; (2018-9-15)
266. Node 2.3: 线性表的链式表示;
267. List 2.3.1: 单链表的定义;
268. 非随机存取; 头指针; 头结点; (2018-9-15)
269. List 2.3.2: 单链表上基本操作的实现;
270. 采用头插法建立单链表; 采用尾插法建立单链表; 按序号查找结点; 按值查找结点; 插入结点操作; 删除结点操作; 求表长操作; (2018-9-15)
271. List 2.3.3: 双链表;
272. 双链表的插入操作; 双链表的删除操作; (2018-9-15)
273. List 2.3.4: 循环链表; 循环双链表; (2018-9-15)
274. List 2.3.4: 循环双链表;
275. List 2.3.5: 静态链表;
276. List 2.3.6: 顺序表和链表的比较;
277. 存取方式; 逻辑结构与物理结构; 插入查找删除; 空间分配; 基于存储的考虑; 基于运算的考虑; 基于环境的考虑; (2018-9-17)
278. Page 37, question 6; end at 16; //////////////////////////////////////////////////////
279. Chapter 3: 栈和队列;
280. Node 3.1: 栈;
281. List 3.1.1: 栈的基本概念;
282. 栈的定义; 栈; 栈顶; 栈底; 空栈; 栈的基本操作; (2018-9-19)
283. List 3.1.2: 栈的顺序存储结构;
284. 顺序栈的实现; 顺序栈的基本运算; 共享栈; (2018-9-19)
285. List 3.1.3: 栈的链式存储结构;
286. List 3.1.4: 试题精选;
287. 选择题11, 卡特兰数; (2018-12-6)
288. Node 3.2: 队列;
289. List 3.2.1: 队列的基本概念;
290. 队列的定义; 队列常见的基本操作; (2018-9-21)
291. Lis 3.2.2: 队列的顺序存储结构;
292. 队列的顺序存储; 循环队列; 牺牲一个单元来区分队空和队满; 循环队列的操作; (2018-9-21)
293. List 3.2.3: 队列的链式存储结构;
294. 队列的链式存储; (2018-9-21)
295. List 3.2.4: 双端队列;
296. List 3.2.5: 试题精选;
297. 练习题8; 练习题10; 练习题18; ;;;;;;;(2018-12-7)
298. Node 3.3: 栈和队列的应用;
299. List 3.3.1: 栈在括号匹配中的应用;
300. List 3.3.2: 栈在表达式求值中的应用;
301. List 3.3.3: 栈在递归中的应用;
302. List 3.3.4: 队列在层次遍历中的应用;
303. List 3.3.5: 队列在计算机系统中的应用;
304. Node 3.4: 特殊矩阵的压缩存储;
305. List 3.4.1: 数组的定义;
306. List 3.4.2: 数组的存储结构;
307. List 3.4.3: 矩阵的压缩存储;
308. 对称矩阵; 三角矩阵; 三对角矩阵; (2018-9-25)
309. List 3.4.4: 稀疏矩阵;
310. List 3.4.5: 本节试题精选;
311. 选择6, 还要再减1; 选择9; (2018-12-8)
312. Chapter 4: 树与二叉树;
313. Node 4.1: 树的基本概念;
314. List 4.1.1: 树的定义;
315. 空树; 根; 子树; (2018-9-26)
316. List 4.1.2: 基本术语;
317. 祖先结点; 子孙结点; 双亲结点; 孩子结点; 兄弟结点; 结点的度; 树的度; 分支结点; 叶子结点; 结点的层次; 结点的深度; 结点的高度; 树的高度; 有序树; 无序树; 路径, 路径长度; 森林; (2018-9-26)
318. List 4.1.3: 树的性质;
319. Node 4.2: 二叉树的概念;
320. List 4.2.1: 二叉树的定义及其主要特性;
321. 二叉树的定义; 几个特殊的二叉树; 满二叉树; 完全二叉树; 二叉排序树; 平衡二叉树; 二叉树的性质; (2018-9-26)
322. List 4.2.2: 二叉树的存储结构;
323. 顺序存储结构; 链式存储结构; (2018-9-27)
324. List 4.2.3: 本节试题精选;
325. 选择14; 选择16; (2018-12-9)
326. Node 4.3: 二叉树的遍历和线索二叉树;
327. List 4.3.1: 二叉树的遍历;
328. 先序遍历; 中序遍历; 后序遍历; 层次遍历; 由遍历序列构造二叉树; (2018-9-28)
329. List 4.3.2: 线索二叉树;
330. 线索二叉树的基本概念; 线索二叉树的构造; 线索二叉树的构造; 线索二叉树的遍历; (2018-9-28)
331. Node 4.4: 树, 森林;
332. List 4.4.1: 树的存储结构;
333. 双亲表示法; 孩子表示法; 孩子兄弟表示法; (2018-10-6)
334. List 4.4.2: 树森林与二叉树的转换;
335. List 4.4.3: 树和森林的遍历;
336. List 4.4.4: 树的应用, 并查集;
337. List 4.4.5: 本节试题精选;
338. 应用题4, 这奇葩的形状; (2018-12-10)
339. Node 4.5: 树与二叉树的应用;
340. List 4.5.1: 二叉排序树;
341. 二叉排序树的定义; 二叉排序树的查找; 二叉排序树的插入; 二叉排序树的构造; 二叉排序树的删除; (2018-10-8)
342. List 4.5.2: 平衡二叉树;
343. 平衡二叉树的定义; 平衡二叉树的插入; 平衡二叉树的查找; (2018-10-8)
344. List 4.5.3: 哈夫曼树和哈夫曼编码;
345. 哈夫曼树的定义; 哈夫曼树的构造; 哈夫曼编码; (2018-10-13)
346. List 4.5.5: 本节试题精选;
347. 选择17, Nh=1+Nh-1+Nh-2; 选择27; 选择29, 把树补全; 解答13, 不要忘了-1; (2018-12-10)
348. Chapter 5: 图;
349. Node 5.1: 图的基本概念;
350. List 5.1.1: 图的定义;
351. 有向图; 无向图; 简单图, 不存在重复边, 不存在顶点到自身的边; 多重图， 允许重边, 允许自环; 完全图, 任意两个顶点之间都存在方向相反的两条弧; 子图; 连通, 连通图和连通分量; 连通分量, 无向图的极大连通子图; 强连通图， 有向图中, 从顶点v到顶点w和从顶点w到顶点v都有路径; 强连通分量; 生成树, 生成森林; 顶点的度, 入度和出度; 边的权和网; 稠密图, 稀疏图; 路径, 路径长度和回路; 简单路径, 简单回路; 距离; 有向树; (2018-10-13)
352. Node 5.2: 图的存储及基本操作;
353. List 5.2.1: 邻接矩阵法;
354. List 5.2.2: 邻接表法;
355. List 5.2.3: 十字链表;
356. List 5.2.4: 邻接多重表;
357. List 5.2.5: 图的基本操作;
358. List 5.2.6: 本节试题精选;
359. 解答题5; (2018-12-11)
360. Node 5.3: 图的遍历;
361. List 5.3.1: 广度优先搜索BFS;
362. BFS算法性能分析; 邻接矩阵O(|V|^2); 邻接表O(|V|+|E|); BFS算法求解单源最短路径问题; 广度优先生成树; 邻接矩阵唯一; 邻接表不唯一; (2018-10-17)
363. List 5.3.2: 深度优先搜索DFS;
364. 类似于先序遍历; DFS算法的性能分析; 深度优先的生成树和生成森林; (2018-10-17)
365. List 5.3.3: 图的遍历与图的连通性;
366. List 5.3.4: 本节试题精选;
367. 选择5, 深度优先回退; 选择6, 深度回退; 选择17, 有向图; (2018-12-11)
368. Node 5.4: 图的应用;
369. List 5.4.1: 最小生成树;
370. 最小生成树不是惟一的; Prim算法; Kruskal算法; (2018-10-24)
371. List 5.4.2: 最短路径;
372. Dijkstra算法求单源最短路径问题; (2018-10-28)
373. List 5.4.3: 拓扑排序;
374. 有向无环图; AOV网; 拓扑排序; (2018-10-28)
375. List 5.4.4: 关键路径;
376. AOE网; 关键路径, 具有最大路径长度的路径; 关键活动; 完成工程的最短时间, 关键路径的长度; 事件vk的最早发生事件;ve(source)=0, ve(k)=max{ve(j)+weight(vj, vk)}; 事件vk的最迟发生时间; vl(dest)=ve(dest), vl(j)=min{vl(k)-weight(vj, vk)}; 活动ai的最早开始时间e(i); e(i)=ve(k); 活动ai的最迟开始时间; l(i)=vl(j)-weight(vk, vj); 一个活动ai的最迟开始时间l(i)和最早开始时间e(i)的差额d(i)=l(i)-e(i); 该活动可以拖延的时间; 关键活动, l(i)-e(i)=0; (2018-11-1)
377. List 5.4.5: 本节试题精选;
378. 应用题11; (2018-12-12)
379. Chapter 6: 查找;
380. Node 6.1: 查找的基本概念;
381. 查找; 查找表(查找结构); 静态查找表, 动态查找表; 关键字; 平均查找长度; (2018-11-1)
382. Node 6.2: 顺序查找和折半查找;
383. List 6.2.1: 顺序查找;
384. 一般线性表的顺序查找; 通过引入哨兵, 可以避免很多不必要的判断语句; 有序表的顺序查找; (2018-11-2)
385. List 6.2.2: 折半查找;
386. 仅适合线性表的顺序存储结构; (2018-11-2)
387. List 6.2.3: 分块查找;
388. List 6.2.4: 本节试题精选;
389. 选择8, ceil(log2(n+1)); 选择15, 别忘了+1-1; (2018-12-13)
390. Node 6.3: B树和B+树;
391. List 6.3.1: B树及其基本操作;
392. B树的高度, 磁盘存取次数; B树的查找; (2018-11-12)
393. List 6.3.3: 本节试题精选;
394. 选择16, 文件系统; 解答5, 查找失败的长度; (2018-12-14)
395. Node 6.4: 散列(Hash)表;
396. List 6.4.1: 散列表的基本概念;
397. 散列函数; 冲突; 同义词; 散列表; (2018-11-12)
398. List 6.4.2: 散列函数的构造方法;
399. 直接定址法; H(key)=a\*key+b; H(key)=key%p; 数字分析法; 数字分析法; 平方取中法; 折叠法; (2018-11-12)
400. List 6.4.2: 处理冲突的方法;
401. 开放定址法; 线性探测法; 平方探测法; 再散列法; 伪随机序列法; 拉链法; (2018-11-12)
402. List 6.4.4: 散列表查找及性能分析;
403. 查找效率取决于散列函数, 处理冲突和装填因子; (2018-11-13)
404. Node 6.5: 字符串模式匹配;
405. List 6.5.1: 简单的模式匹配算法;
406. List 6.5.2: 改进的模式匹配算法--KMP算法;
407. Chapter 7: 排序;
408. Node 7.1: 排序的基本概念;
409. List 7.1.1: 排序的定义;
410. 排序; 算法的稳定性; 比较和移动; (2018-11-14)
411. Node 7.2: 插入排序;
412. List 7.2.1: 直接插入排序;
413. 稳定的排序方法; 适用于顺序存储和链式存储的顺序表; (2018-11-14)
414. List 7.2.2: 折半插入排序;
415. 稳定的排序方法; (2018-11-14)
416. List 7.2.3: 希尔排序;
417. 不稳定; (2018-11-14)
418. Node 7.3: 交换排序;
419. List 7.3.1: 冒泡排序;
420. 最小的元素如气泡一般逐渐往上漂浮直到水面; 稳定的排序算法; (2018-11-15)
421. List 7.3.2: 快速排序;
422. 不稳定; (2018-11-15)
423. Node 7.4: 选择排序;
424. 每一趟选出关键字最小的元素; (2018-11-16)
425. List 7.4.1: 简单选择排序;
426. 简单选择排序是一个不稳定的排序方法; (2018-11-16)
427. List 7.4.2: 堆排序;
428. 小根堆; 大根堆; 交换n和n/2; 练习题1, 实在搞不定就行特殊值; (2018-11-18)
429. Node 7.5: 归并排序和基数排序;
430. List 7.5.1: 归并排序;
431. 稳定; (2018-11-19)
432. List 7.5.2: 基数排序;
433. 不是基于比较进行排序的; 稳定; (2018-11-18)
434. Node 7.6: 各种内部排序算法的比较及应用;
435. List 7.6.1: 内部排序算法的比较;
436. List 7.6.2: 内部排序算法的应用;
437. Node 7.7: 外部排序;
438. List 7.7.1: 外部排序的基本概念;
439. List 7.7.2: 外部排序的方法;
440. List 7.7.3: 多路归并与败者树;
441. List 7.7.4: 置换-选择排序(生成初始归并段);
442. List 7.7.5: 最佳归并树;
443. ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
444. 2018真题
445. 填空1, 注意是循环; 填空5, 注意A[4][5]含义; 填空8, 递增顺序; ;;;
446. -