

数据结构算法

1.1 数组和字符串 2

- 1.1.1 一维数组的倒置 2
  - 范例1-1 一维数组的倒置 2
  - 相关函数: fun函数
- 1.1.2 一维数组应用 3
  - 范例1-2 一维数组应用 3
- 1.1.3 一维数组的高级应用 5
  - 范例1-3 一维数组的高级应用 5
- 1.1.4 显示杨辉三角 7
  - 范例1-4 显示杨辉三角 7
  - 相关函数: c函数 8
- 1.1.5 魔方阵 9
  - 范例1-5 魔方阵 9
  - 相关函数: InitArray函数
- 1.1.6 三维数组的表示 14
  - 范例1-6 三维数组的表示 14
- 1.1.7 多项式的数组表示 17
  - 范例1-7 多项式数组的表示 17
- 1.1.8 查找矩阵的马鞍点 19
  - 范例1-8 查找矩阵的马鞍点 19
  - 相关函数: Get\_Saddle函数
- 1.1.9 对角矩阵建立 21
  - 相关函数: Store函数
  - 范例1-9 对角矩阵建立 21
- 1.1.10 三对角矩阵的建立 22
  - 范例1-10 三对角矩阵的建立 22
  - 相关函数: Store函数
- 1.1.11 三角矩阵建立 24
  - 相关函数: Store函数
  - 范例1-11 三角矩阵建立 24
- 1.1.12 对称矩阵的建立 25
  - 范例1-12 对称矩阵的建立 25
  - 相关函数: store函数
- 1.1.13 字符串长度的计算 28
  - 范例1-13 字符串长度的计算 28
  - 相关函数: strlen函数
- 1.1.14 字符串的复制 29
  - 范例1-14 字符串的复制 29
  - 相关函数: strcpy函数
- 1.1.15 字符串的替换 31
  - 范例1-15 字符串的替换 31
  - 相关函数: strep函数
- 1.1.16 字符串的删除 33
  - 范例1-16 字符串的删除 33
  - 相关函数: strdel函数
- 1.1.17 字符串的比较 35
  - 范例1-17 字符串的比较 35
  - 相关函数: strcmp函数
- 1.1.18 字符串的抽取 36
  - 范例1-18 字符串的抽取 36
  - 相关函数: substr函数
- 1.1.19 字符串的分割 38
  - 范例1-19 字符串的分割 38
  - 相关函数: partition函数
- 1.1.20 字符串的插入 40
  - 范例1-20 字符串的插入 40
  - 相关函数: insert函数
- 1.1.21 字符串的匹配 42
  - 范例1-21 字符串的匹配 42
  - 相关函数: nfind函数
- 1.1.22 字符串的合并 43
  - 范例1-22 字符串的合并 43
  - 相关函数: catstr函数
- 1.1.23 文本编辑 45
  - 范例1-23 文本编辑 45
  - 相关函数: StrAssign函数

1.2 栈和队列 54

- 1.2.1 用数组仿真堆栈 54
  - 范例1-24 用数组仿真堆栈 54
  - 相关函数: push函数 pop函数
- 1.2.2 用链表仿真堆栈 57
  - 范例1-25 用链表仿真堆栈 57
  - 相关函数: push函数 pop函数
- 1.2.3 顺序栈公用 59
  - 范例1-26 顺序栈公用 59
  - 相关函数: push函数 pop函数
- 1.2.4 进制转换问题 61
  - 范例1-27 进制转换问题 61
  - 相关函数: MultiBaseOutput函数
- 1.2.5 顺序队列操作 64
  - 范例1-28 顺序队列操作 64
  - 相关函数: push函数 pop函数
- 1.2.6 循环队列 66
  - 范例1-29 循环队列 66
  - 相关函数: EnQueue函数 DeQueue函数
- 1.2.7 链队列的入队、出队 69
  - 范例1-30 链队列入队、出队 69
  - 相关函数: push函数 pop函数
- 1.2.8 舞伴问题 71
  - 范例1-31 舞伴问题 71
  - 相关函数: EnQueue函数 DeQueue函数 DancePartner函数

1.3 链表 75

- 1.3.1 头插法建立单链表 75
  - 范例1-32 头插法建立单链表 75
  - 相关函数: createlist函数
- 1.3.2 限制链表长度建立单链表 77
  - 范例1-33 限制链表长度建立单链表 77
  - 相关函数: createlist函数
- 1.3.3 尾插法建立单链表 79
  - 范例1-34 尾插法建立单链表 79
  - 相关函数: createlist函数
- 1.3.4 按序号查找单链表 80
  - 范例1-35 按序号查找单链表 80
  - 相关函数: getnode函数
- 1.3.5 按值查找单链表 82
  - 范例1-36 按值查找单链表 82
  - 相关函数: locatenode函数
- 1.3.6 链表的插入 84
  - 范例1-37 链表的插入 84
  - 相关函数: insertnode函数
- 1.3.7 链表的删除 86
  - 范例1-38 链表的删除 86
  - 相关函数: deletelist函数
- 1.3.8 归并两个单链表 88
  - 范例1-39 归并两个单链表 88
  - 相关函数: concatenate函数
- 1.3.9 动态堆栈 90
  - 范例1-40 动态堆栈 90
  - 相关函数: push函数 Pop函数
- 1.3.10 动态队列 93
  - 范例1-41 动态队列 93
  - 相关函数: Enqueue函数
- 1.3.11 初始化单循环链表 95
  - 范例1-42 初始化单循环链表 95
  - 相关函数: ListLength\_CL函数
- 1.3.12 查询元素的前驱和后继 98
  - 范例1-43 查询元素的前驱和后继 98
  - 相关函数: PriorElem\_CL函数 NextElem\_CL函数
- 1.3.13 单循环链表中元素的删除 101
  - 范例1-44 单循环链表中元素的删除 101
  - 相关函数: ListDelete\_CL函数
- 1.3.14 单循环链表的清除和销毁 107
  - 范例1-45 单循环链表的清除和销毁 107
  - 相关函数: DestroyList函数
- 1.3.15 仅设表尾指针循环链表的合并 110
  - 范例1-46 仅设表尾指针循环链表的合并 110
  - 相关函数: MergeList\_CL函数
- 1.3.16 正序输出双向链表 113
  - 范例1-47 正序输出双向链表 113
  - 相关函数: ListInsert函数 ListTraverse函数
- 1.3.17 逆向输出双向链表 116
  - 范例1-48 三角矩阵建立 116
  - 相关函数: ListTraverseBack函数
- 1.3.18 删除双向链表中的节点 121
  - 范例1-49 删除双向链表中的节点 121
  - 相关函数: ListDelete函数
- 1.3.19 双向链表的元素个数 124
  - 范例1-50 双向链表的元素个数 124
  - 相关函数: ListLength函数
- 1.3.20 判断双向链表是否为空 126
  - 范例1-51 判断双向链表是否为空 126
  - 相关函数: ListEmpty函数
- 1.3.21 双向链表元素值的查询 129
  - 范例1-52 双向链表元素值的查询 129
  - 相关函数: GetElemP函数
- 1.3.22 稀疏矩阵的建立 136
  - 范例1-53 稀疏矩阵的建立 136
  - 相关函数: Create函数
- 1.3.23 稀疏矩阵的删除 138
  - 范例1-54 稀疏矩阵的删除 138
  - 相关函数: erase函数

1.4 树和二叉树 141

- 1.4.1 获得二叉树的深度和根 (顺序结构) 141
  - 范例1-55 获得二叉树的深度和根 141
  - 相关函数: BiTreeDepth函数 Root函数
- 1.4.2 获得二叉树的深度和根 (链表结构) 144
  - 范例1-56 获得二叉树的深度和根 144
  - 相关函数: BiTreeDepth函数 Root函数
- 1.4.3 树的插入 (顺序结构) 147
  - 相关函数: InsertChild函数
  - 范例1-57 树的插入 147
- 1.4.4 节点的修改 (顺序结构) 150
  - 范例1-58 节点的修改 150
  - 相关函数: Assign函数
- 1.4.5 节点的修改 (链式结构) 154
  - 相关函数: Assign函数
  - 范例1-59 节点的修改 154
- 1.4.6 双亲、孩子和兄弟节点的查询 (顺序结构) 158
  - 范例1-60 双亲、孩子和兄弟节点的查询 158
  - 相关函数: Parent函数 LeftChild函数 RightChild函数 LeftSibling函数 RightSibling函数
- 1.4.7 双亲、孩子和兄弟节点的查询 (链式结构) 162
  - 范例1-61 双亲、孩子和兄弟节点的查询 162
  - 相关函数: Parent函数 LeftChild函数 RightChild函数 LeftSibling函数 RightSibling函数
- 1.4.8 中序遍历二叉树 (顺序结构) 169
  - 范例1-62 中序遍历二叉树 169
  - 相关函数: InOrderTraverse函数
- 1.4.9 中序遍历二叉树 (链式结构) 171
  - 相关函数: InOrderTraverse函数
  - 范例1-63 中序遍历二叉树 171
- 1.4.10 中序非递归遍历二叉树 (链式结构) (1) 174
  - 范例1-64 中序非递归遍历二叉树 174
  - 相关函数: InOrderTraverse函数
- 1.4.11 中序非递归遍历二叉树 (链式结构) (2) 177
  - 范例1-65 中序非递归遍历二叉树 177
  - 相关函数: InOrderTraverse2函数
- 1.4.12 后序遍历二叉树 (顺序结构) 180
  - 范例1-66 后序遍历二叉树 180
  - 相关函数: PostOrderTraverse函数
- 1.4.13 后序遍历二叉树 (链式结构) 183
  - 相关函数: PostOrderTraverse函数
  - 范例1-67 后序遍历二叉树 183
- 1.4.14 层次遍历二叉树 (顺序结构) 186
  - 范例1-68 层次遍历二叉树 186
  - 相关函数: LevelOrderTraverse函数
- 1.4.15 层次遍历二叉树 (链式结构) 188
  - 范例1-68 层次遍历二叉树 188
  - 相关函数: LevelOrderTraverse函数
- 1.4.16 树的合并 191
  - 范例1-70 树的合并 191
  - 相关函数: Find函数 Union函数
- 1.4.17 树的二叉链表存储的基本操作 193
  - 相关函数: LevelOrderTraverse函数
  - 范例1-71 树的二叉链表存储的基本操作 193
- 1.4.18 二叉树的三叉链表存储的基本操作 201
  - 范例1-72 二叉树的三叉链表存储表示 201
  - 相关函数: CreateBiTree函数
- 1.4.19 二叉树的二叉线索存储的基本操作 212
  - 相关函数: CreateBiThrTree函数
  - 范例1-73 二叉树的二叉线索存储 212
- 1.4.20 树的双亲表存储的基本操作 215
  - 相关函数: CreateTree函数
  - 范例1-74 树的双亲表存储的基本操作 215
- 1.4.21 哈夫曼编码 (1) 223
  - 相关函数: HuffmanCoding函数
  - 范例1-75 哈夫曼编码 (1) 223
- 1.4.22 哈夫曼编码 (2) 226
  - 范例1-76 哈夫曼编码 (2) 226
  - 相关函数: HuffmanCoding函数

1.5 排序 229

- 1.5.1 直接插入排序 229
  - 范例1-77 直接插入排序 229
  - 相关函数: InsertSort函数
- 1.5.2 折半插入排序 (顺序结构) 231
  - 相关函数: BInsertSort函数
  - 范例1-78 折半插入排序 (顺序结构) 231
- 1.5.3 2—路插入排序 (顺序结构) 233
  - 范例1-79 2—路插入排序 (顺序结构) 233
  - 相关函数: P2\_InsertSort函数
- 1.5.4 折半插入排序 (链式结构) 235
  - 范例1-80 折半插入排序 (链式结构) 235
  - 相关函数: Arrange函数
- 1.5.5 2—路插入排序 (链式结构) 238
  - 相关函数: Rearrange函数
  - 范例1-81 2—路插入排序 (链式结构) 238
- 1.5.6 希尔排序 241
  - 范例1-82 希尔排序 241
  - 相关函数: ShellSort函数
- 1.5.7 冒泡排序 243
  - 范例1-83 冒泡排序 243
  - 相关函数: bubble\_sort函数
- 1.5.8 一趟快速排序 246
  - 范例1-84 一趟快速排序 246
  - 相关函数: QSort函数
- 1.5.9 一趟快速排序的改进算法 248
  - 范例1-85 一趟快速排序的改进算法 248
  - 相关函数: QuickSort函数
- 1.5.10 简单选择排序 250
  - 范例1-86 简单选择排序 250
  - 相关函数: SelectSort函数
- 1.5.11 箱子排序 252
  - 范例1-87 箱子排序 252
  - 相关函数: sort函数
- 1.5.12 树型选择排序 254
  - 范例1-88 树型选择排序 254
  - 相关函数: TreeSort函数
- 1.5.13 堆排序 256
  - 相关函数: HeapSort函数
  - 范例1-89 堆排序 256
- 1.5.14 归并排序 258
  - 范例1-90 归并排序 258
  - 相关函数: MergeSort函数
- 1.5.15 多路平衡归并排序 260
  - 范例1-91 多路平衡归并排序 260
  - 相关函数: K\_Merge函数
- 1.5.16 置换—选择排序 265
  - 相关函数: Replace\_Selection函数
  - 范例1-92 置换—选择排序 265
- 1.5.17 文件的归并 269
  - 范例1-93 文件的归并 269
  - 相关函数: K\_Merge函数

1.6 查找 272

- 1.6.1 顺序表的查找 273
  - 范例1-94 顺序表的查找 273
  - 相关函数: Search\_Seq函数
- 1.6.2 静态树表的查找 276
  - 范例1-95 静态树表的查找 276
  - 相关函数: Search\_SOSTree函数
- 1.6.3 二叉排序树的基本操作 280
  - 范例1-96 二叉排序树的基本操作 280
  - 相关函数: InsertBST函数
- 1.6.4 平衡二叉树的基本操作 285
  - 相关函数: SearchBST函数
  - 范例1-97 平衡二叉树的基本操作 285
- 1.6.5 B树的基本操作 290
  - 相关函数: SearchBTree函数
  - 范例1-98 B树的基本操作 290
- 1.6.6 按关键字字符串的遍历双键树 295
  - 相关函数: SearchDLTree函数
  - 范例1-99 按关键字字符串遍历双键树 295
- 1.6.7 按关键字字符串的遍历Trie树 301
  - 范例1-100 按关键字字符串遍历Trie树 301
  - 相关函数: SearchTrie函数
- 1.6.8 哈希表的基本操作 306
  - 范例1-101 哈希表的基本操作 306
  - 相关函数: SearchHash函数

1.7 图 311

- 1.7.1 图的邻接矩阵存储表示 311
  - 范例1-102 图的邻接矩阵存储表示
  - 相关函数: CreateFAG函数 CreateDG函数
- 1.7.2 图的邻接表存储表示 324
  - 相关函数: CreateFAG函数
  - 范例1-103 图的邻接表存储表示 324
- 1.7.3 有向图的十字链表存储表示 335
  - 范例1-104 有向图的十字链表存储表示 335
  - 相关函数: CreatedG函数
- 1.7.4 无向图的邻接多重表存储表示 344
  - 相关函数: CreateGraph函数
  - 范例1-105 无向图的邻接多重表存储表示 344
- 1.7.5 最小生成树 355
  - 范例1-106 最小生成树 355
  - 相关函数: MiniSpanTree\_PRIM函数
- 1.7.6 关节点和重连通分量 359
  - 范例1-107 关节点和重连通分量 359
  - 相关函数: FindArticul函数
- 1.7.7 拓扑排序 366
  - 范例1-108 拓扑排序 366
  - 相关函数: TopologicalSort函数
- 1.7.8 关键路径 374
  - 范例1-109 关键路径 374
  - 相关函数: CriticalPath函数
- 1.7.9 最短路径 383
  - 范例1-110 最短路径 383
  - 相关函数: ShortestPath\_DIJ函数
- 1.7.10 每一对顶点之间的最短路径 387
  - 相关函数: ShortestPath\_FLOYD函数
  - 范例1-111 每一对顶点之间的最短路径 387