**2020112921 刘欣豪 ( 结果统一放在最后 )**

第一次作业

作业说明：

1. 截止日期为 2022 年 12 月 xx 号（最后一周）;
2. 以纸质版的形式提交给助教;
3. 请遵守学术和道德规范，严禁抄袭。出现雷同作业时，抄袭者与被抄袭者均扣分。

# 小作业

### 选择题

1. 数据结构是指 (A ) 的集合以及它们之间的 (B )。
2. A. 数据元素 B. 计算方法 C. 逻辑存储 D. 数据映像
3. A. 结构 B. 关系 C. 运算 D. 算法
4. 计算机处理的数据一般具备某种内在联系，这是指 (B )。

A. 数据和数据之间存在某种联系 B. 元素和元素之间存在某种联系 C. 元素内部具有某种结构 D. 数据项和数据项之间存在某种联系

1. 在数据结构中，与所使用计算机无关的是数据的 (A ) 结构。

A. 逻辑 B. 存储 C. 物理 D. 以上都对

1. 计算机的算法是指解决某一问题的有限运算序列，它必须具备输入、输出和 (B )。

A. 可行性、可移植性和可扩充性 B. 可行性、有穷性和确定性 C. 确定性、有穷性和稳定性

D. 易读性、稳定性和确定性

1. 某算法的时间复杂度是n2，表明该算法的 ( C)。

A. 问题规模是n2 B. 执行时间等于n2 C. 执行时间正比于n2 D. 问题规模正比于n2

1. 以下函数中，时间复杂度最低的是 (A )

A. 1() = 1000 log2 B. 2() = log2 − 1000 log C. 3() = −

2

2

## 1000 log2 D. 4() = 2 log2 − 1000 log2

1. 在长度为 n 的顺序表中插入一个元素的时间复杂度为 (C )，删除一个元素的时间复杂度为 (C )

## A. (1) B. (log2 ) C. () D. ( )

2

1. 使用二路归并算法，将两个长度分别为 , 的递增有序顺序表合并成一个递增有序顺序表，最多比较次数为 ( B)

## A. B. + − 1 C. + D. 不确定

1. 带头结点的单链表 为空的判断条件是 (B )

## A. L is null B. head.next==None C. head.next=L D. L is not null

1. 在双链表中，删除结点 p 之后的一个结点（非尾部结点）的语句是 (D )

## p.next=p.next.next p.next.next.prior=p

1. p.next.prior=p p.next=p.next.next

## p.next=p.next.next p.next.prior=p

1. p.next.next=p.next p.next.prior=p

### 判断题

* 1. 算法的好坏与算法的描述语言无关，但是受到所使用的计算机影响；F
  2. 线性表中每个元素都有一个前驱元素与后继元素；F
  3. 在单链表中，可以从头结点开始查找任意一个结点；T
  4. 在循环单链表中，可以从任一结点出发，通过前后移动遍历整个表；T
  5. 所谓顺序栈，就是指元素按照大小顺序入栈形成的栈；F

### 填空题

1. 在顺序栈中，如果已经进栈 − 1 个元素，将第 个元素进栈时，进栈算法的时间复杂度为O(1);

2.

使用二路归并算法，将两个长度分别为 , 的递增有序顺序表合并成一个递增有序顺序表，最小比较次数为MIN(n,m);

3.

对于一个具有 ( ≥ 1) 个 i 九二点的单链表，在结点 后插入一个新结点的时间复杂度是O(1)，在第一个值为 的结点之后插入新结点的时间复杂度为O(n);

4.

在一个双链表 中，若要在结点 （非头结点）之前插入一个新结点 ，需要执行的操作是q.prior=p.prior,q.prior.next=q,q.next=p,p.prior=q;

5.

两个栈共享一个存储区，利用 data[1, … , ] 表示其中的元素，栈 1 在左侧（小下标） 处，栈 2 在右侧（大下标）处。两个栈顶指针分别为 1 与 2，则当栈 1 为空时

1 =1, 栈 2 为空时 2 =n, 栈满时top1-top2=1;

### 简答题

1. 当设计解决某一问题的算法时，需要选择合适的数据结构，需要从哪些方面考虑？
2. 顺序表和链表各有什么优劣，应该如何选择？

3. 设 个元素进栈序列时 (1, 2, … , )，其输出的序列是 1, 2, 3 … , ，如果 2 = 3， 试分析 的可能取值情况。

4. 1

分析以下算法的时间复杂度

**def** fun(n): i=1

**while** i<=n: i=i\*2

# return i

5. 分析以下算法的时间复杂度

**def** fun(n): s=0

**for** i **in** range(n):

**for** j **in** range(i):

**for** k **in** range(j): s++

# return s

### 编程题

1. 有一个循环单链表，设计一个算法删除其最后一个最大值的结点；
2. 给定一个字符串 ，设计一个算法采用顺序栈判断 是否是满足形如“序列 1@ 序列 2”的字符串，其中“序列 2”为“序列 1”的逆序， 中仅存在 1 个 @ 符号。

# 大作业

在教科书中，我们学习到了括号的匹配，即形如“( )(( )){([( )])}”，“((( )(( )){([( )])}))”等为正确的括号，而“)(( )){([( )])}”，“({[ ])}”为不正确的符号关系。本次作业我们将类似的思想应用到标记语言的匹配。常见的标记语言有 html 和 xml，其中 html 是网络超文本文档的标准格式，而 xml 常用于各种数据集。例如，交通领域常用的宏观仿真软件 MATSIM 的输入和输出数据即是以 xml 格式储存。下图对比了 html 与对应的文本。

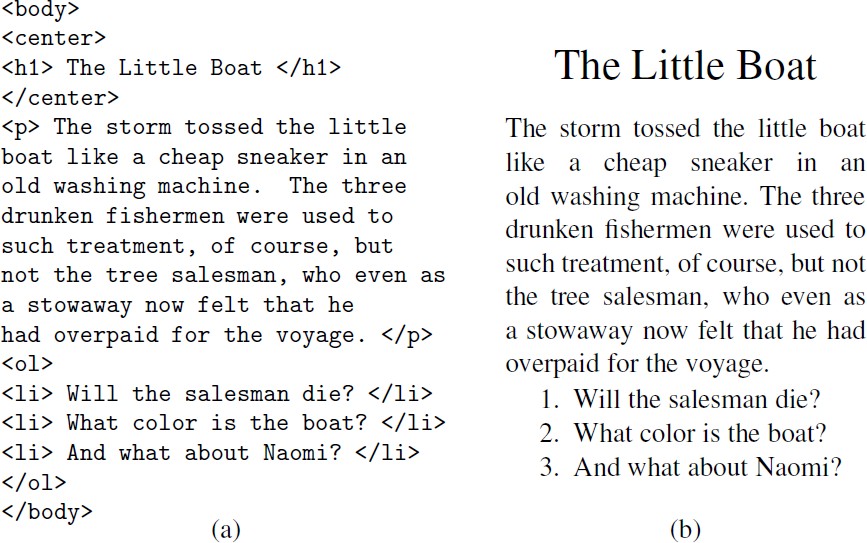


Figure 1: 左图为 html，右图为文本

在 html 中，部分文本由其标签分割，一个简单的 html 文本块开始于 < 标签名 >（称之为“开始标签”），结束于 < /标签名 >（称之为“结束标签”）。例如图中的首行的 < body> 与尾行的 < /body> 是一对。在图中还用到了如下标签：

* body: 文档的主体
* h1：章节的标题
* center：居中
* p：段落
* ol：编号命令列表
* li：编号项

尽管浏览器允许存在某些错误匹配的标签，我们还是希望 html 中存在的不匹配问题越少约好。请使用基于栈，设计一个判断 html 中标签是否匹配的算法。

提示：python 的字符串 str 可以使用 str.find(sstr, a) 方法检测字符串是否包含子字符串

，其中 a 表示开始的位置，即查找 a 到字符串尾部是否存在 sstr。该函数的返回值为所发现的第一个满足条件的 sstr 在 str 中的开始位置。

思路：我们先建立一个存储开始标签的栈 S，从字符串开头寻找”<“，然后从此处出发寻找“>”，获取二者之间的子字符串，如果不为空也不存在“/”，则说明是一个开始标签，放入栈 中；否则，则从 S 出栈一个标签，判断与其是否对应。然后从当前位置继续向下寻找”<“， 循环以上步骤，直至文本结尾。