

## 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 ( ) 的集合以及它们之间的 ( )。  
(1) A. 数据元素 B. 计算方法 C. 逻辑存储 D. 数据映像  
(2) A. 结构 B. 关系 C. 运算 D. 算法
2. 计算机处理的数据一般具备某种内在联系, 这指的是 ( )。  
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
3. 在数据结构中, 与所使用计算机无关的是数据的 ( ) 结构。  
A. 逻辑 B. 存储 C. 物理 D. 以上都对
4. 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 ( )。  
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
5. 某算法的时间复杂度是  $n^2$ , 表明该算法的 ( )。  
A. 问题规模是  $n^2$  B. 执行时间等于  $n^2$  C. 执行时间正比于  $n^2$  D. 问题规模正比于  $n^2$
6. 以下函数中, 时间复杂度最低的是 ( )  
A.  $T_1(n) = 1000 \log_2 n$  B.  $T_2(n) = n^{\log_2 n} - 1000 \log_2 n$  C.  $T_3(n) = n^2 - 1000 \log_2 n$  D.  $T_4(n) = 2n \log_2 n - 1000 \log_2 n$
7. 在长度为  $n$  的顺序表中插入一个元素的时间复杂度为 ( ), 删除一个元素的时间复杂度为 ( )  
A.  $O(1)$  B.  $O(\log_2 n)$  C.  $O(n)$  D.  $O(n^2)$
8. 使用二路归并算法, 将两个长度分别为  $n, m$  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 ( )  
A.  $n$  B.  $n + m - 1$  C.  $n + m$  D. 不确定
9. 带头结点的单链表  $L$  为空的判断条件是 ( )  
A.  $L$  is null B.  $\text{head.next} == \text{None}$  C.  $\text{head.next} = L$  D.  $L$  is not null

10. 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（ ）
- A.  $p.next=p.next.next$   $p.next.next.prior=p$
  - B.  $p.next.prior=p$   $p.next=p.next.next$
  - C.  $p.next=p.next.next$   $p.next.prior=p$
  - D.  $p.next.next=p.next$   $p.next.prior=p$

#### 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；
2. 线性表中每个元素都有一个前驱元素与后继元素；
3. 在单链表中，可以从头结点开始查找任意一个结点；
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；
5. 所谓顺序栈，就是指元素按照大小顺序入栈形成的栈；

#### 填空题

1. 在顺序栈中，如果已经进栈  $i-1$  个元素，将第  $i$  个元素进栈时，进栈算法的时间复杂度为 \_\_\_\_\_；
2. 使用二路归并算法，将两个长度分别为  $n, m$  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 \_\_\_\_\_；
3. 对于一个具有  $n(n \geq 1)$  个结点的单链表，在结点  $p$  后插入一个新结点的时间复杂度是 \_\_\_\_\_，在第一个值为  $x$  的结点之后插入新结点的时间复杂度为 \_\_\_\_\_；
4. 在一个双链表  $L$  中，若要在结点  $p$ （非头结点）之前插入一个新结点  $q$ ，需要执行的操作是 \_\_\_\_\_，\_\_\_\_\_，\_\_\_\_\_，\_\_\_\_\_；
5. 两个栈共享一个存储区，利用  $data[1, \dots, n]$  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  $top1$  与  $top2$ ，则当栈 1 为空时  $top1 = \underline{\hspace{1cm}}$ ，栈 2 为空时  $top2 = \underline{\hspace{1cm}}$ ，栈满时 \_\_\_\_\_；

#### 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  $n$  个元素进栈序列时  $(p_1, p_2, \dots, p_n)$ ，其输出的序列是  $1, 2, 3, \dots, n$ ，如果  $p_2 = 3$ ，试分析  $p_1$  的可能取值情况。
4. 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 *str*，设计一个算法采用顺序栈判断 *str* 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，*str* 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。

```

<body>
<center>
<h1> The Little Boat </h1>
</center>
<p> The storm tossed the little
boat like a cheap sneaker in an
old washing machine. The three
drunken fishermen were used to
such treatment, of course, but
not the tree salesman, who even as
a stowaway now felt that he
had overpaid for the voyage. </p>
<ol>
<li> Will the salesman die? </li>
<li> What color is the boat? </li>
<li> And what about Naomi? </li>
</ol>
</body>

```

(a)

## The Little Boat

The storm tossed the little boat like a cheap sneaker in an old washing machine. The three drunken fishermen were used to such treatment, of course, but not the tree salesman, who even as a stowaway now felt that he had overpaid for the voyage.

1. Will the salesman die?
2. What color is the boat?
3. And what about Naomi?

(b)

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越好。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 *sstr*，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。