

武汉大学数学与统计学院2017-2018学年第一学期期末考试
数据结构与算法 (A卷)

姓名: _____ 学号: _____

1. (20分) Python相关

(a) (5分) 字典与集合

i. 下表给出了某个人的一些信息，以此创建一个字典：

Name	Li Ming
Sex	Male
Hometown	Wuhan
Age	22

ii. 创建一个集合，其内容为2, 'Python', True, 3.14。

(b) (5分) 仔细阅读以下程序，写出运行结果：

```
alist = list(range(20))
print(alist[:10])
print(alist[-10:])
print(alist[:10:2])
print(alist[::2])
```

(c) (5分) 将以下两段代码改成用列表生成式(list comprehension)实现：

```
alist = []
for x in range(1, 11):
    if x % 2 == 0:
        alist.append(x * x)
```

```
alist = []
for m in 'abc':
    for n in '1234':
        alist.append(m + n)
```

(d) (5分) 给定长 (length) 和宽 (width)，编写长方形 (Rectangle) 的类，其中包括求面积 (area) 和周长 (circum) 的方法。

2. (15分) 线性表的链表实现

(a) (6分) 编写结点类：

```
class Node(object):
    def __init__(self, data):
        self.data = data
        self.next = None
    def getData(self):
        return self.data
    def getNext(self):
        ...
    def setData(self, newdata):
        ...
    def setNext(self, newnext):
        ...
```

(b) (9分) 编写无序表类：

```
class UnorderedList(object):
    def __init__(self):
        self.head = None

    def add(self, item): //将新结点插入当前链表的表头，即头结点的后面
        temp = ...
        temp.setNext(...)
        ...

    def search(self, item): //在链表中查找某个元素是否存在
        current = self.head
        found = False
        while ...:
            if ...:
                found = True
            else:
                ...
        return found

    def remove(self, item): //在链表中删除某个元素
        current = self.head
        previous = None
        found = False
        while not found:
            if current.getData() == item:
                found = True
            else:
                ...
                ...
        if previous == None:
            ...
        else:
            previous.setNext(current.getNext())
```

3. (15分) 栈

(a) (6分) 补充以下代码，实现栈类（假定列表右侧为栈顶）：

```
class Stack(object):
    def __init__(self):
        self.items = []
    def isEmpty(self):
        return self.items == []
    def push(self, item): //入栈
        ...
    def pop(self): //出栈
        ...
    def peek(self): //返回栈顶元素，但不删除
        ...
```

(b) (9分) 编写函数，利用栈类实现十进制到二进制的转换：

```
def dec2bin(decNumber):
    stack = Stack()
    while decNumber > 0:
        rem = decNumber % 2
        ...
```

```

        decNumber = decNumber // 2
    binString = ""
    while ...:
        ...
    return binString

```

4. (20分) 队列与二叉树

(a) (4分) 编写队列类 (假定列表左侧为队尾) :

```

class Queue(object):
    def __init__(self):
        self.items = []
    def isEmpty(self):
        return self.items == []
    def enqueue(self, item): //入队列
        ...
    def dequeue(self): //出队列
        ...

```

(b) (4分) 利用结点-引用方式编写二叉树类:

```

class BinaryTree(object):
    def __init__(self, data):
        self.data = data
        self.lchild = None
        self.rchild = None
    def getLeftChild(self):
        return self.lchild
    def getRightChild(self):
        return self.rchild
    def insertLeft(self, newdata): //插入左孩子
        if self.lchild == None:
            ...
        else:
            ...
            ...
            ...
    //插入右孩子, 与插入左孩子类似, 此处忽略细节, 以下问题中可直接使用
    def insertRight(self, newdata):
        ***

```

(c) (4分) 编写中序遍历函数:

```

def inorder(tree):
    if ...:
        ...
        ...
        ...

```

(d) (4分) 利用队列类编写层序遍历函数:

```

def levelorder(tree):
    if tree == None:
        return
    q = Queue()
    node = tree

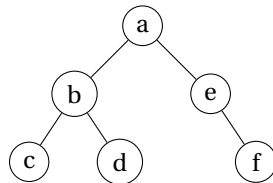
```

```

...
while not q.isEmpty():
    ...
    print(node.data)
    if node.lchild != None
        ...
    if node.rchild != None:
        ...

```

(e) (4分) 给定二叉树



利用以上类和函数，编写程序构造该二叉树，实现其层序遍历和中序遍历，并写出运行结果。

5. (15分) 给定一串数 53, 24, 91, 88, 34, 71, 44, 18,

(a) (5分) 编写无序列表的顺序查找函数：

```

def sequentialSearch(alist, item):
    pos = 0
    found = False
    ...
    return found

```

(b) (5分) 以图示+说明的方式阐述无序列表的归并排序算法，或直接编写归并排序函数：

```

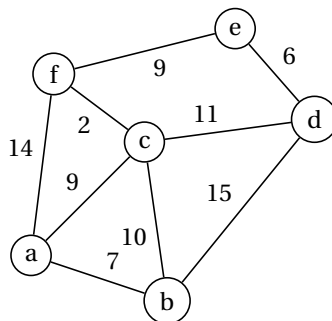
def mergeSort(alist):
    ...

```

(c) (5分) 调用上述函数，

- 根据这串数构造一个列表；
- 查找88是否在该列表中，并分别输出相关信息；
- 输出原列表，对其进行排序，再输出新列表的结果。

6. (15分) 给定无向图



(a) (5分) 写出邻接矩阵；

(b) (4分) 以字典的形式，写出每个顶点的度；

(c) (6分) 根据该邻接矩阵，写出从顶点 a 出发，深度优先和广度优先搜索的结果。