北京大学考场纪律

1、考生要按规定的考试时间提前 5 分钟进入考场，隔位就座或按照监考人员的安排就座，将学生证放在桌面。无学生证者不能参加考试；迟到超过 15 分钟不得入场；与考试无关人员不得进入考场。如考试允许提前交卷，考生在考试开始 30 分钟后可交卷离场；未交卷擅自离开考场，不得重新进入考场继续答卷；交卷后应离开考场，不得在考场内逗留或在考场附近高声喧哗。

2、除非主考教师另有规定，学生只能携带必要的文具参加考试，其它所有物品（包括空白纸张、手机和智能手表等电子设备）不得带入座位；已经带入考场的手机和智能手表等电子设备必须关机，并与其他物品一起集中放在监考人员指定位置，不得随身携带或带入座位及旁边。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放和收回，考生不得带出考场。考生在规定时间前答完试卷，应举手示意请监考人员收卷后方可离开；答题时间结束监考人员宣布收卷时，考生应立即停止答卷，在座位上等待监考人员收卷清点无误后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准旁窥、交头接耳、打暗号或做手势，不准携带与考试课程内容相关的材料，不准携带具有发送、接收信息功能或存储有与考试课程内容相关材料的电子设备（如手机、智能手表、非教师允许的计算器等），不准抄袭或协助他人抄袭试题答案或者与考试课程内容相关的资料，不准窃取、索要、强拿、传、接或者交换试卷、答卷、草稿纸或其他物品，不准代替他人或让他人代替自己参加考试，等等。凡违反考试纪律或作弊者，按《北京大学本科考试工作与学习纪律管理规定》给予相应处分。

5、考生须确认填写的个人信息真实、准确，并承担信息填写错误带来的一切责任

诚信宣言：

我承诺，严格遵守校规校纪，诚信考试！

考生签名： _____

北京大学信息学院考试试卷

考试科目：_____**姓名：**_____ **学号：**_____.
考试时间： 2021 年___月___日 下午　　**任课教师：**胡俊峰

题号	一	二	三	四	五	六	七	八	**总分**
分数									
阅卷人									

一、 **多重选择题：（每小题 2 分，共 16 分）**
从候选答案中选择 1 到多个正确答案，每个选项占 0.5 分

2. Python 对象减少引用计数的情况有：
A）离开当前的名字空间
B）从包含该对象的容器中移除
C）名字被 del 操作删除
D）赋值操作
答案：（　　）

3、在机器学习中，PCA 可以有以下那些作用：

A）对数据完成中心化标准化

B）对数据特征实现正交化

C）对数据特征进行降维

D）过滤小强度随机噪声
答案：（　　）

4、常见的将非平稳序列转换成平稳序列的方法有：

A)　差分

B)　对原序列做回归

C)　对原序列平方

D)　对原序列将原序列进行滑动平均

答案：（　　）

5、 关于图像处理，以下选项说法正确的是：
A) 灰度直方图均衡化可以增加图片的对比度
B) SIFT 特征具有旋转不变性
C) HoG 特征具有旋转不变性
D) Sobel 算子用于边缘检测
答案：（ ）

7、以下说法中正确的有：
A) 协程之间是并行运行的
B) 朴素贝叶斯分类器属于生成式模型
C) 隐马尔可夫模型中，状态序列与观察值序列都是马尔科夫链
D) 样本空间中某方向上的方差越大，说明其信息熵越高
答案：（ ）

8、下列关于 PyTorch 或神经网络的相关说法，哪些是正确的？
A) 在 loss.backward()一步中，对网络的参数进行了更新
B) ReLU 是线性函数，所以只使用 ReLU 作为激活函数无法训练出非线性的网络
C) 模型过拟合时可采取数据增强的方法
D) RNN 模型大小不会因为输入增加而增加
答案：（ ）

二、阅读程序并给出运行结果（共 30 分，每小题 3 分）

Python 语言部分：
4、

```
import copy
lst = ['a', 'b', ['c', 'd']]
lst3 = copy.deepcopy(lst)
lst3[0] = 'e'
lst3[2][0] = 'f'
lst3.append('g')
lst3[2].append('h')
print(lst)
print(lst3)
```

请写出上面程序的运行结果：

5、

```
def func_a(func_a_arg_a, func, **kwargs):
    print(func_a_arg_a)
    func(**kwargs)

def func_b(arg_a):
    print(arg_a)

def func_c():
    print('Hello World')

if __name__ == '__main__':
    func_a(func_a_arg_a='temp', arg_a='Hello Python', func=func_b)
    func_a(func_a_arg_a='temp', func=func_c)
```

请写出上面程序的运行结果：

6、

```
def fun(items):
    se = set()
    for it in items:
        if it not in se:
            yield it
            se.add(it)
a = [1, 5, 2, 1, 9, 1, 5, 10]
print(list(fun(a)))
```

请写出上面程序的运行结果：

7、

```
mat_1 = [['a','b','c'],['d','e','f']]
mat_2 = ['1','2','3']
result = ','.join([i+j
    for vec in mat_1
    for i,j in zip(vec, mat_2)])
print(result)
```

请写出上面程序的运行结果：

8、

```
def call_Fun_counter(func):
    def wrapper(*args, **kwargs):
        wrapper.calls += 1
        return func(*args, **kwargs)
    wrapper.calls = 0
    return wrapper

@call_Fun_counter
def fib(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fib(n-1) + fib(n-2)

print(fib(4))
print(fib.calls)
```

请写出上面程序的运行结果：

9、

```
import numpy as np
a = np.array([[1,2,3],[2,3,4]])
b = a
b += 1
print(a)
a = a.T + b[0,1:]
print(a)
```

请写出上面程序的运行结果：

10、

```
import numpy as np
import pandas as pd
df=pd.DataFrame([[1,np.nan,2],
                 [2,0,5],
                 [np.nan,4,6],
                 [3,4,0]],columns=list('ABC'))
df_2=pd.DataFrame([[0,3],[1,5]],columns=list('BE'))
display(df,df_2)

df_3=df.fillna(0)
df_3['D']=df_3['C'].apply(lambda x:x**2)
display(df_3)
display(pd.merge(df_3,df_2,on='B',how='inner'))
```

df:

	A	B	C
0	1	NaN	2
1	2	0	5
2	NaN	4	6
3	3	4	0

df2:

	B	E
0	0	3
1	1	5

请根据上述程序写出最后两个display语句的运行结果（分别计分）

三、填空（共 16 分）

前 4 题代码填空部分要求用 python 完成填空

(如果一行能完成，尽量写一行，多于一行代码视情况可能会扣 0.5-1 分)

1、请用列表生成式生成 50 以内 4 的倍数（2 分）

_____

3、标准化是特征处理的常规方法，对于给定的矩阵 X，请按列对其进行标准化（x-mean/std）（2 分）

```
X = np.random.random((100, 30))
```

X_mean = _____(1 分)

X_std = _____(1 分)

4.补全下列任务代码（共 4 分）

HITS(2 分)：给定 M 以及已初始化的 H、A,请补全迭代更新 H2、A2 的代码

```
import numpy as np
while True:
    _____
    _____
    A2/=np.linalg.norm(A2)
    H2/=np.linalg.norm(H2)
    if np.allclose(H,H2) and np.allclose(A,A2):#判断向量是否相近
       break
    A,H=A2,H2
```

PageRank（2 分）：给定 M 以及已初始化的 PR，请补全迭代更新 PR2 的代码

```
import numpy as np
d=0.85
while True:
    _____
    if np.allclose(PR2,PR):
        break
```

5、熵在信息论中是很重要的概念。熵度量了一个编码的信息量。对于随机变量 x，它的所有可能取值为 $x=\{x_1,x_2,..,x_n\}$，概率密度函数为 $p(x)$. 则熵为

$H(x) = -\sum_{x_i} p(x_i)\log(x_i)$。当概率密度函数是均匀分布时，不确定性最大，同时熵值也达到最大，为 _____

6、考虑 CNN 的卷积操作，输入为 长 L*宽 L*通道数 c，卷积核大小为 长 k*宽 k，共有 m 个卷积核，填充为 p 步长 s，问输出层的尺寸（size）和通道数分别为？（3 分）

_____， _____

五、简答题（共 12 分）：

1. TCP 协议的英文或中文全称是什么？（1 分）
   请简述 socket 通讯中服务器端与客户端进行握手实现通讯连接的基本过程（3 分）（可以用 python 代码或伪代码或文字来说明）

2、简述协程概念以及 python 中有哪两种实现协程的机制？（4 分）

六、python 函数实现：（共 16 分）

1、下面给出了一个二叉树的类型定义

```
class BinaryTree(object):
    def __init__(self,rootObj):
        self.key = rootObj
        self.leftChild = None
        self.rightChild = None

    def insertLeft(self,newNode):
        if self.leftChild == None:
            self.leftChild = BinaryTree(newNode)
        else:
            t = BinaryTree(newNode)
            t.leftChild = self.leftChild
            self.leftChild = t

    def insertRight(self,newNode):
        if self.rightChild == None:
            self.rightChild = BinaryTree(newNode)
        else:
            t = BinaryTree(newNode)
            t.rightChild = self.rightChild
            self.rightChild = t

    def getRightChild(self):
        return self.rightChild

    def getLeftChild(self):
        return self.leftChild
    def setRootVal(self,obj):
        self.key = obj

    def getRootVal(self):
         return self.key
```

要求：

1）写出语句序列生成一个该类型的实例 r，包含 3 个结点，根节点内容为字符串“+“，左子树节点内容为字符串“15”，右子树内容为字符串“10”（2 分）

语句序列：

2）为这个 BinaryTree 类添加一个成员函数 countLeaf 方法，实现对实例中节点数的计数，并返回计数值。比如上面那个树的实例，调用该方法返回值为 3（2 分）
语句序列（包含函数定义和添加成员函数到类中的语句）：

3、 请补充代码完成基于概率图的短语划分动归算法：

```
from collections import defaultdict

## 词的最大长度
MAX_LEN=5

def Segmentation(word_sequence, normalized_frequency,
quality_estimator):
    ## Input:
    ##  word_sequence:  需要进行分词的单词序列
    ##  normalized_frequency(dict): 单词的频次，用来评价短语质量，本次
考试可以忽略
    ##  quality_estimator(dict): 短语质量评估，词典或函数，可以返回候选
短语的评分值
    ## Output:
    ## result:返回分词结果

    token_cnt = len(word_sequence)

    ## 初始化
    ## h 记录当前最优划分分值
    ## g 记录当前最优划分结果
    h=[-1]*(token_cnt+1)  # 用-1 初始化，长度 token_cnt+1，n+1 个间隔
    g=[0]*(token_cnt+1)   # 用 0 初始化 ...

    h[0] = 1

    for l_idx in range(token_cnt):
        for phrase_len in range(1, MAX_LEN+1):
            r_idx = l_idx + phrase_len - 1
            tmp_phrase = word_sequence[l_idx:r_idx+1]
            if r_idx >= token_cnt:
#完成序列加工
                break
```

```
            else:
    ### TODO：计算当前的分值，判断是否需要更新最优划分，如果需要则更新 h 记录当前最优的分值，更新 g 来记录当前最优短语划分方案（4 分）



    ### FINISH

    ##  计算并返回最优短语划分结果
    l_idx = token_cnt
    result = []
    while l_idx>0:
    ### TODO：根据前面的计算结果，生成划分后的短语列表（2 分）



    ### FINISH
    result.reverse()
    return result
```