5.

class A:

p = 1

class B(A):

pass

class C(A):

pass

1) print A.p, B.p, C.p

显示：1 1 1

2)

B.p = 2

print A.p, B.p, C.p

显示：1 2 1

注意：该代码是python2的。

BQXX7431.jpg

(8) 下列程序执行后的输出结果为：

x = 1; y = 2; z = 'a';

if x << 1 > 2:

print "X"

elif x == 97:

print "Z"

else:

print "Y"

A、X B、Y C、Z D、出错

选 B，

注意，该代码是python2

二、不定项选择

(1) 下面的函数转换哪一个是正确的（语法均正确）：

A. int("ABcDef");

B. float('5')

C. bool((3, 5, 7))

D. str(5)

选 BCD

(2)

三、填空题

(1) 以下函数需要在其中引入一个全局变量k，请填写语句：

def func():

\_\_\_\_\_\_\_\_\_\_\_ global k

k = k + 1;

(2) 请把以下函数转化为Python Lambda匿名函数：

def add(x, y):

return x+y

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ add = lambda x, y: x + y

4. 写一个函数，给定矩阵的长度级数n，返回一个回旋排列的数字矩阵。

例如：

n=2返回：

1 2

4 3

n=3返回：

1 2 3

8 9 4

7 6 5

n=4返回：

1 2 3 4

12 13 14 5

11 16 15 6

10 9 8 7

DNJB6571.jpg

2. 表达式 [1]\*2 的值为（C）

A. [2]

B. [1]

C. [1,1]

D. [1,2]

3. 已知列表x=[1,2,3]，那么执行语句x.insert(1,4)只有，x的值为（B）

A. [1,2,3,4]

B. [1,4,2,3]

C. [4,3,2,1]

D. [4,2,3]

4. 已知字符串x='hello world'，那么执行语句x.replace('hello','hi')之后，x的值为（C）

A. 'hello world'

B. 'hillo world'

C. 'hi world'

D. ' world'

DOUW1999.jpg

... （一个html文档）

要求爬取所有的公告名称，公告时间。编写程序解析段。

3. 简述Scrapy的框架（可做图说明），及各模块的基本功能，如果没有使用过Scrapy 框架，那么挑选一个你使用过的框架进行说明。

4. 请实现如下功能（不限语言）：给定一个无序的数值序列，找出其中第N大的数值。

最平常的思路是将数组排序，最快的排序是快排，然后返回已排序数组的第k个数，算法时间复杂度为O（nlogn），空间复杂度为O（1）。

使用快排的思想，但是每次只对patition之后的数组的一半递归，这样可以将时间复杂度将为O（n）。

5. 二选一：

5.1 开发问题：如何衡量两段文本的相识度？

5.2 开发问题：当你值爬取数据，遇到反爬时，解决思路是什么？

FGEL4894.jpg

要求：

1. 不允许写伪代码，要求实现一个能运行的完整函数或方法

2. 不要使用任何语言或第三方类库自带的方法

必答题：

0. 定义一个哈系表的成员和方法

下面两题二选一：

1. 写出一个方法输出1～100之间所有的素数么（素数定义为在大于1的自然数中，除了1和它本身以外不再有其他因数，比如2,3,5都是素数，4,6都不是素数）

2. 计整数二进制表示中1的个数

下面两题二选一：

3. 给定一个整数数组，实现快速排序算法进行升序排序，比如：[2,5,8,9,3]排序后得到：[2,3,5,8,9]

4. 一个楼梯有N个台阶，小明从台阶最底层地面上楼梯，小明一次可最大跨3阶（也就是说每次迈步可以上1阶，2阶，或者3阶）。问小明爬上顶一共有多少中步伐组合？

FMTN1281.jpg

三、算法题：

有两个特别大的手机号码文件，A文件包含100万个号码（每行1个手机号），B文件包含1000万个号码，请找出A，B文件中共有的手机号码。

（从算法和数据结构角度来思考，大数据量有性能要求，写清解决问题思路即可，可以不写详细代码）

GDZI2530.jpg

1. 写出20个linux命令

login, logout, passwd, who, su, sudo

date, cal, echo, alias, bg, fg, cd, dirs, eval, exec, uname,

ls,

ps, kill,

history,

less, more, cat,

man, info, help,

source, export, ln, unlink, unset, set,

2.

4.

>>> ================================== pass

5.

GYKR3955.jpg

4. 3+4j不是合法的Python表达式。（错）

5. Python集合可以包含相同的元素。（错）

6. 在UTF-8编码中一个汉字需要占用3个字节。（错）

注意：

UTF-8 是变长的，1-6个字节。

少数是汉字每个占用3个字节，多数占用4个字节。

占用3个字节的汉字： 52156 个

占用4个字节的汉字： 64029 个

7. Python 3.x 支持使用中文作为变量名。（对）

8. 函数中必须包含return语句。（错）

9. 栈和队列的都具有先入后出的特点。（错）

10. 定义类时实现类\_\_eq\_\_()方法，该类对象即可支持运算符==。（对）

三、

HOPD1716.jpg

一、单项选择题

(1) 以下叙述正确的是：d

a. continue语句的作用是结束整个循环的执行

b. 只能在循环体内和switch语句体内使用break语句

c. 在循环体内使用break语句或continue语句的作用相同

d. 从多层循环嵌套中退出时，只能使用goto语句

(2) Python如何定义一个函数：c

a. class <name>(<Type> arg1, <Type> arg2, ..., <Type> argN)

b. function <name>(arg1, arg2, ..., argN)

c. def <name>(arg1, arg2, ..., argN)

d. def <name>(<Type> arg1, <Type> arg2, ..., <Type> argN)

(3) 下面哪个函数能够在linux环境下创建一个子进程：b

a. os.popen

b. os.fork

c. os.system

d. os.link

(4) 已知x=43, ch='A', y=1，则表达式（x>=y and ch<'B' and y）的值是：b

a. 0

b. 1

c. 出错

d. True("真")

(5) 下面的语句哪个会无限循环下去：b

其中range函数原型为range([start], stop[, step])

a.

for a in range(10):

time.sleep(10)

b.

while 1 < 10:

time.sleep(10)

c.

while True:

break

d.

a = [3,-1,5,7]

for i in a[:]:

if a > 10:

break

(6) 下列表达式中返回为True的是：

a. 3 > 2 > 2

b. 'abc' > 'xyz'

c. 0x56 < 56

d. (3,2) < ('a','b')

注意：

python2中，d的返回是True。

python3中，全部返回False。

(7) Python不支持的数据类型有：a

a. char

b. int

c. float

d. list

IBWF1980.jpg

>>> ================================== pass

2. 平衡点问题：

假如一个数组中的元素，其前面的部分等于后面的部分，那么这个点的位序就是平衡点。

比如一个列表 numbers =

[1,3,5,7,8,25,4,20]。25前面的总和为24，25后面的总和也是24，那么这个点就是平衡点。

要求编写程序，寻找并返回任意一个列表的平衡点。

JMUQ1263.jpg

一个html表单源码。

KEQO0150.jpg

10.

2. 下面代码执行后的结果：

def f(x, l=[]):

for i in range(x):

l.append(i\*i)

print l

>>> f(2)

[0, 1]

>>> f(3, [3,2,1])

[3, 2, 1, 0, 1, 4]

>>> f(3)

[0, 1, 0, 1, 4]

5. \_\_new\_\_和\_\_init\_\_的区别

\_\_init\_\_是当实例对象创建完成后被调用的，然后设置对象属性的一些初始值。

\_\_new\_\_是在实例创建之前被调用的，因为它的任务就是创建实例然后返回该实例，是个静态方法。

10. 有两个序列a, b，大小都为n，序列元素的值任意整型数，无序。

要求：通过交换a,b中的元素，使[序列a元素的和]与[序列b元素的和]之间的差最小。

求解思路：

A = sum(a) - sum(b)

a[i]和b[j]进行交换后：

A' = sum(a) - a[i] + b[j] - (sum(b) - b[j] + a[i])

= sum(a) - sum(b) - 2 \* (a[i] - b[j])

= A - 2 \* (a[i] - b[j])

设 x = a[i] - b[j]，则

A' = A - 2x

x = (A - A') / 2

设 0 < A，则

1) x = (0..A)

x = A / 2，效果最好。

2) x 不属于(0..A)

则当前的a或者b，就是答案。

所以算法如下：

在a和b中寻找使得x=(0..A)，并且最接近A/2的i和j，交换相应的i和j元素。

重新计算A后，重复前面的步骤，直到找不到(0..A)之间的x为止。

算法：

1）将2个序列合并为一个序列，并排序。为序列Source

2）拿出最大元素Big，次大元素Small。

3）在余下的序列Source[:-2]进行平分，得到序列max, min

4）将Small加到max序列，将Big加到min序列，重新计算序列和，和大的为max，小的为min。

def swap\_balance(arr\_a, arr\_b):

''' 通过交换a,b中的元素，使[序列a元素的和]与[序列b元素的和]之间的差最小 '''

N = len(arr\_a)

sum\_a = sum(arr\_a)

sum\_b = sum(arr\_b)

# 计算序列和

diff = sum\_a - sum\_b

while diff != 0:

best\_i = best\_j = best\_change = 0

for i in range(N):

for j in range(N):

change = arr\_a[i] - arr\_b[j]

if abs(diff - 2 \* change) < abs(diff - 2 \* best\_change):

best\_change = change

best\_i = i

best\_j = j

if best\_change == 0:

return False

arr\_a[best\_i], arr\_b[best\_j] = arr\_b[best\_j], arr\_a[best\_i]

# 重新计算序列和

sum\_a -= best\_change

sum\_b += best\_change

diff = sum\_a - sum\_b

return true

12. 如何列出表'test'内name域值不为'tecmint'，web\_address域值为'test.com'的所有数据？

select \*

from test

where name != 'tecmint'

and web\_address = 'test.com'

;

3.

15. 如何使用标准库pdb调试Python程序

1) 启动

$ python -m pdb mymodule.py

or

>>> import pdb

>>> import mymodule

>>> pdb.run('mymodule.test()')

2) 设置断点

pdb.set\_trace()

3) 常用调试命令

h(elp) [command]

无参数时，打印可用命令的列表。使用命令作为参数，打印有关该命令的帮助。

w(here)

打印堆栈跟踪，最新的帧在底部。

d(own) [count]

将当前帧计数（默认一个）向下移动到堆栈跟踪（更新的帧）。

u(p) [count]

将当前帧计数（默认值）移动到堆栈跟踪（到较早的帧）。

b(reak) [([filename:]lineno | function) [, condition]]

使用lineno参数，在当前文件中设置断点。

使用函数参数，在该函数中的第一个可执行语句中设置断点。

行号可以用文件名和冒号作为前缀，以在另一个文件（可能是尚未加载的文件）中指定断点。

在sys.path上搜索文件。

请注意，每个断点都分配有一个数字，所有其他断点命令都引用该数字。

如果存在第二个参数，它是一个表达式，在断点被执行之前必须计算为true。

无参数，列出所有中断，包括每个断点，断点被命中的次数，当前忽略计数和相关条件（如果有）。

tbreak [([filename:]lineno | function) [, condition]]

临时断点，当它第一次被击中时被自动删除。参数与break相同。

cl(ear) [filename:lineno | bpnumber [bpnumber ...]]

使用filename：lineno参数，清除此行的所有断点。用断点号的空格分隔列表，清除那些断点。没有论证，清除所有休息（但首先要求确认）。

disable [bpnumber [bpnumber ...]]

禁用以空格分隔的断点号列表形式给出的断点。禁用断点意​​味着它不能导致程序停止执行，但是与清除断点不同，它保留在断点列表中，并且可以（重新）启用。

enable [bpnumber [bpnumber ...]]

启用指定的断点。

ignore bpnumber [count]

设置给定断点号的忽略计数。如果省略count，则忽略计数设置为0。当忽略计数为零时，断点变为活动状态。当非零时，每次达到断点时计数递减，并且不禁用断点，并且任何关联条件的计算结果为true。

condition bpnumber [condition]

为断点设置新的条件，这是在断点被执行之前必须计算为true的表达式。如果条件不存在，则删除任何现有条件；即断点是无条件的。

commands [bpnumber]

指定断点号bpnumber的命令列表。命令本身显示在以下行。键入仅包含end的行以终止命令。

s(tep)

执行当前行，在第一个可能的时间停止（在调用的函数中或当前函数中的下一行）。

n(ext)

继续执行直到到达当前函数中的下一行或返回。（next和step之间的区别是step在被调用函数内停止，而next （几乎）全速，只在当前功能的下一行停止。）

unt(il) [lineno]

如果没有参数，继续执行直到到达具有大于当前数字的数字的行。

使用行号，继续执行，直到到达具有大于或等于的数字的行。在这两种情况下，当当前帧返回时也停止。

在版本3.2中更改：允许给出明确的行号。

r(eturn)

继续执行直到当前函数返回。

c(ont(inue))

继续执行，只有在遇到断点时停止。

j(ump) lineno

设置将要执行的下一行。仅在最下面的框架中可用。这允许您跳回并再次执行代码，或者向前跳转以跳过您不想运行的代码。

应该注意的是，不是所有的跳转都被允许 - 对于实例，不可能跳到for循环中或从finally子句。

l(ist) [first[, last]]

列出当前文件的源代码。如果没有参数，请列出当前行周围的11行或继续上一个列表。使用.作为参数，列出当前行周围的11行。使用一个参数，在该行列出11行。有两个参数，列出给定的范围；如果第二个参数小于第一个，它将被解释为计数。

当前帧中的当前行由->指示。如果正在调试异常，则异常最初提出或传播的行由>>指示，如果它与当前行不同。

版本3.2中的新功能： >>标记。

ll | longlist

列出当前函数或框架的所有源代码。有趣的行标记为list。

a(rgs)

打印当前函数的参数列表。

p expression

计算当前上下文中的表达式并打印其值。

注意

print()也可以使用，但不是调试器命令 - 这将执行Python print()函数。

pp expression

像p命令，除了表达式的值是使用pprint模块打印的。

whatis expression

打印表达式的类型。

source expression

尝试获取给定对象的源代码并显示它。

display [expression]

如果表达式更改，则每次在当前帧中停止执行时，显示表达式的值。

不带表达式时，列出当前帧的所有显示表达式。

undisplay [expression]

不要在当前框架中再显示表达式。如果没有表达式，请清除当前框架的所有显示表达式。

interact

启动一个交互式解释器（使用code模块），其全局命名空间包含当前范围中找到的所有（全局和局部）名称。

alias [name [command]]

创建一个名为name的执行 命令 的别名。The command must not be enclosed in quotes. 可替换参数可由%1，%2等指示，而%\*由所有参数替换。如果未给出命令，则会显示名称的当前别名。如果没有给出参数，则列出所有别名。

别名可以嵌套，并且可以包含可以在pdb提示符下合法输入的任何内容。请注意，内部pdb命令可以被别名覆盖。然后隐藏这样的命令，直到去除别名。别名被递归地应用于命令行的第一个字；所有其他单词在行中都是孤独的。

unalias name

删除指定的别名。

！ statement

在当前堆栈帧的上下文中执行（一行）语句。可以省略感叹号，除非语句的第一个单词类似于调试器命令。要设置全局变量，您可以在同一行上使用global语句为赋值命令添加前缀。

run [args ...]

restart [args ...]

重新启动调试的Python程序。如果提供了参数，则使用shlex进行分割，并将结果用作新的sys.argv。历史，断点，动作和调试器选项被保留。restart是run的别名。

q(uit)

从调试器中退出。正在执行的程序中止。

>>> ================================== pass

17. 如何提高爬取效率

18.

20. 如何爬取一个千亿级数据量查询系统的内容（每个查询都带有验证码功能），

请写出你的整体思路？

1) 购买验证码库api

2) 分布式 + 多进程 + 协程 + ip代理

7. 请写出可以一定程度判断网页是否变化的响应头。

1) Last-Modified / If-Modified-Since

在浏览器第一次请求某一个URL时，服务器端的返回状态会是200，内容是你请求的资源，同时有一个Last-Modified的属性标记此文件在服务期端最后被修改的时间。

Last-Modified: Fri, 12 May 2006 18:53:33 GMT

客户端第二次请求此URL时，根据 HTTP 协议的规定，浏览器会向服务器传送 If-Modified-Since 报头，询问该时间之后文件是否有被修改过。

If-Modified-Since: Fri, 12 May 2006 18:53:33 GMT

如果服务器端的资源没有变化，则自动返回 HTTP 304 （Not Changed.）状态码，内容为空，这样就节省了传输数据量。

当服务器端代码发生改变或者重启服务器时，则重新发出资源，返回和第一次请求时类似。

从而保证不向客户端重复发出资源，也保证当服务器有变化时，客户端能够得到最新的资源

2) ETag / If-None-Match

HTTP 协议规格说明定义ETag为“被请求变量的实体值”。另一种说法是，ETag是一个可以与Web资源关联的记号（token）。典型的Web资源可以一个Web页，但也可能是JSON或XML文档。服务器单独负责判断记号是什么及其含义，并在HTTP响应头中将其传送到客户端，以下是服务器端返回的格式：

ETag: “50b1c1d4f775c61:df3”

客户端的查询更新格式是这样的：

If-None-Match: W/”50b1c1d4f775c61:df3”

如果ETag没改变，则返回状态304然后不返回，这也和Last-Modified一样。

PXJE7884.jpg

In machine learning, naive Bayes classifiers are a family of simple probabilistic

classifiers based on applying Bayes' theorem with strong (naive) independence

assumptions between the features.

Naive Bayes has been studied extensively since the 1950s. It was introduced under

a different name into the text retrieval community in the early 1960s, and remains

a popular (baseline) method for text categorization, the problem of judging

documents as belonging to one category or the other (such as spam or legitimate,

sports or politics, etc.) with word frequencies as the features. With appropriate

pre-processing, it is competitive in this domain with more advanced methods

including support vector machines. It also finds application in automatic medical

diagnosis.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear

in the number of variables (features/predictors) in a learning problem. Maximum-

likelihood training can be done by evaluating a closed-form expression, with

takes linear time, rather than by expensive iterative approximation as used for

many other types of classifiers.

In the statistics and computer science literature, Naive Bayes models are known

under a variety of names, including simple Bayes and independence Bayes. All

these names reference the use of Bayes' theorem in the classifier's decision rule,

but naive Bayes is not (necessarily) a Bayesian method.

1. 请翻译该段文字

https://zh.wikipedia.org/wiki/朴素贝叶斯分类器

在机器学习中，单纯贝氏分类器 是一系列以假设特征之间强（朴素）独立下运用贝叶斯定理为基础的简单概率分类器。

单纯贝氏自20世纪50年代已广泛研究。

在20世纪60年代初就以另外一个名称引入到文本信息检索界中，并仍然是文本分类的一种热门（基准）方法，文本分类是以词频为特征判断文件所属类别或其他（如垃圾邮件、合法性、体育或政治等等）的问题。

通过适当的预处理，它可以与这个领域更先进的方法（包括支持向量机）相竞争。它在自动医疗诊断中也有应用。

单纯贝氏分类器 是高度可扩展的，因此需要数量与学习问题中的变量（特征/预测器）成线性关系的参数。

最大似然训练可以通过评估一个封闭形式的表达式来完成，只需花费线性时间，而不需要其他很多类型的分类器所使用的费时的迭代逼近。

在统计学和计算机科学文献中，单纯贝氏模型有各种名称，包括简单贝叶斯和独立贝叶斯。

所有这些名称都参考了贝叶斯定理在该分类器的决策规则中的使用，但单纯贝氏不（一定）用到贝叶斯方法。

2. 如了解该算法，请写出算法对应的伪码

三、编程题目

htQRKV1905.jpg

06. 场景：一个大于内存的txt日志文件，如果统计出某个字符出现的次数，请简述。

文件以字符串模式打开，分块(256MB)依次读入内存。

对每块数据，统计该数据块中，某个字符出现的次数。

str.count(s) 统计字符串中，字符s的数量

07. 场景：现有sever1, server2, server3三台消费者，生产者1通过api接口来调用，也就是说三台消费者都部署类api访问，请简述如何优化负载均衡。

方法一：

使用nginx，对3台消费者做负载均衡。

方法二：

使用celery + rabbitmq，通过异步消息队列，对生产者做缓冲。

当消费者处理完当前请求后，会从rabbitmq上获取新的任务。

08. 场景：股票行情源1不稳定，住要表现在获取所有股票行情的时候，有些股票会意外中断如timeout，请问如何优化程序。

分块获取行情数据，每块返回100只股票的数据。

如果还会中断，继续减少每块的数据量。

09. 场景：股票行情获取两种方法：

1) 全量获取

2) 增量获取

现需获取前复权数据（如当天有复权，历史行情会变化），全量获取速度会比增量获取慢。

请问如何以最快速度获取股票行情。

回答：

1) 当天复权的股票，使用全量获取数据。

2) 其它股票，使用增量获取数据。

10. 场景：根据字符串来筛选股票，并将股票从CSV中读取导入到程序中，并返回结果。简述如何更快的返回结果。

使用正则表达式，筛选股票。

使用pandas库，从csv文件中读取数据。

10. 以下的代码的输出将是什么？说出你的答案并解释

class Parent(object):

x = 1

class Child1(Parent):

pass

class Child2(Parent):

pass

print Parent.x, Child1.x, Child2.x

Child1.x = 2

print Parent.x, Child1.x, Child2.x

Parent.x = 3

print Parent.x, Child1.x, Child2.x

输出：

1 1 1

1 2 1

3 2 3

解释：

1 2 1

因为Child1.x = 2

3 2 3

因为Parent.x =

13. 在函数中使用global关键字声明全局变量

>>> def func():

>>> global COUNT

>>> pass

14.

15. 补充下面函数代码

def deco(func):

补充代码部分

@deco

def myfunc2(a,b):

print(" myfunc(%s, %s) called." % (a, b))

return a+b

回答：

def deco(func):

def wrapper(\*args, \*\*kwargs):

ret = func(\*args, \*\*kwargs)

return wrapper

07.

13. 利用正则表达式从一段html文本中提取多个字段值

>>> import re

>>>

>>> content = '''

>>> <td>

>>> <a href="https://www.baidu.com/articles/zj.html" title="浙江省">浙江省主题介绍</a>

>>> <a href="https://www.baidu.com//articles/gz.html" title="贵州省">贵州省主题介绍</a>

>>> </td>

>>> '''

>>>

>>> # 获取<a href></a>之间的内容

>>> print('获取链接文本内容:')

获取链接文本内容:

>>> res = r'<a .\*?>(.\*?)</a>'

>>> m = re.findall(res, content, re.S|re.M)

>>> for value in m:

... print(value)

...

浙江省主题介绍

贵州省主题介绍

>>>

>>> # 获取所有<a href></a>链接所有内容

>>> print('获取完整链接内容:')

获取完整链接内容:

>>> urls = re.findall(r"<a.\*?href=.\*?<\/a>", content, re.I|re.S|re.M)

>>> for i in urls:

... print(i)

...

<a href="https://www.baidu.com/articles/zj.html" title="浙江省">浙江省主题介绍</a>

<a href="https://www.baidu.com//articles/gz.html" title="贵州省">贵州省主题介绍</a>

>>>

>>> # 获取<a href></a>中的URL

>>> print('获取链接中URL:')

获取链接中URL:

>>> res\_url = r"(?<=href=\").+?(?=\")|(?<=href=\').+?(?=\')"

>>> link = re.findall(res\_url, content, re.I|re.S|re.M)

>>> for url in link:

... print(url)

...

https://www.baidu.com/articles/zj.html

https://www.baidu.com//articles/gz.html

15. python如何定义一个函数？

def 函数名(传入函数的值):

pass

16.

17.

18. 任意一种排序方法的代码实现（尽量选择时间复杂度是 O(NlogN) ）

链接：https://blog.csdn.net/ester\_yc/article/details/77933182

时间复杂度 空间复杂度

类别 排序方法 ----------------------------------------------------------------稳定性

平均情况 最好 最坏 辅助存储

------------------------------------------------------------------------------------------

插入 直接插入 O(n\*\*2) O(n) O(n\*\*2) O(1) 稳定

shell排序 O(n\*\*1.3) O(n) O(n\*\*2) O(1) 不稳定

选择 直接排序 O(n\*\*2) O(n\*\*2) O(n\*\*2) O(1) 不稳定

堆排序 O(n\*log2(n)) O(n\*log2(n)) O(n\*log2(n)) O(1) 不稳定

交换 冒泡排序 O(n\*\*2) O(n) O(n\*\*2) O(1) 稳定

快速排序 O(n\*log2(n)) O(n\*log2(n)) O(n\*\*2) O(n\*log2(n)) 不稳定

归并排序 O(n\*log2(n)) O(n\*log2(n)) O(n\*log2(n)) O(1) 稳定

基数排序 O(d(r+n)) O(d(n+rd)) O(d(r+n)) O(rd+n) 稳定

归并排序的代码:

def merge(left, right):

i, j = 0, 0

result = []

while i < len(left) and j < len(right):

if left[i] <= right[j]:

result.append(left[i])

i += 1

else:

result.append(right[j])

j += 1

result += left[i:]

result += right[j:]

return result

def merge\_sort(lists):

# 归并排序

if len(lists) <= 1:

return lists

num = len(lists) / 2

left = merge\_sort(lists[:num])

right = merge\_sort(lists[num:])

return merge(left, right)

三、编程题（任选一题）

>>> ================================== pass

1.

2.

SWZT5862.jpg

二、编程题（选用你最熟悉的语言如java, javascript, python, c++）

1. 有这样几个变量：

char name[16]="小涛";

int order=2;

float score=90.5;

char result[128];

请把name, order, score这几个变量按格式拼接，放到result里：

"小涛同学考试成绩为90.5，全班排第 2 名"

（注意变量类型转换，字符串拼接，或者用语言提供的特殊方法）

>>> fmt = '{name}同学考试成绩为{score}，全班排第 {order} 名'

>>> result = fmt.format(name=name, score=score, order=order)

3.

8.

TRIZ2642.jpg

链接：https://www.cnblogs.com/lilin567/p/7417095.html

1. 填空题

(01) 获取随机数的方法：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Math.random

(02) JS中将一个变量强制改为浮点类型的方法：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

parseFloat(String)

(03) 根据id获取元素的原生js方法：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

getElementById();

(04) 将文档声明为HTML5文档类型，需要在文档头添加：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<!doctype html>

(05) Chrome浏览器的私有样式前缀：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-webkit-

(06) CSS3 设置圆角的属性名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

border-radius

(07) CSS3 中，#是根据\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_来选择元素，.是根据\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_来选择元素。

id, class

(08) form标签特有的属性请列举三个：\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_。

action, method, enctype, target

(09) 正则表达式中，$是什么意识：\_\_\_\_\_\_\_\_

从字符串的末尾开始从后向前匹配

(10) typeof'123'返回：\_\_\_\_\_\_\_\_

number

2. 简答题

(1) 自定义一个浏览器兼容的事件绑定方法需要注意哪些问题？

对以存在的事件绑定方法进行判断。

能兼容各主流浏览器。

IE中没有捕获阶段，只能处理冒泡阶段。

(2)

(3

(4) display和position都有哪些值？分别列举一下这些值的用途。

display：block/inline/inline-block /table/table-cell/none

position: relative/absolute/fixed/static

display:block;/\*当前元素以块级形式显示，默认宽度为父元素，可设定宽高，换行显示\*/

display:inline;/\*当前元素以行内形式显示，默认宽度为内容宽度，不可设宽高，同行显示\*/

display:inline-block;/\*显示时，默认宽度为内容宽度，可设宽高，同行显示\*/

display:table;/\*显示规则和block相同，但是样式为table的样式\*/

display:table-cell;/\*以table 单元格的样式显示\*/

display:none;/\*元素小时，位置也不占\*/

position:relative;/\*当前元素被设为相对定位，元素在文档流中，百分比的参照物是元素本身\*/

position:absolute;/\*当前元素被设为绝对定位，元素脱离文档流，定位参照物：第一个定位祖先/根元素\*/

position:fixed;/\*当前元素被设为固定定位，默认宽度为内容宽度，脱离文档流，参照物是视窗\*/

(5) 地址栏输入一个地址后，输入回车，描述一下这时开始，浏览器做类什么。

同 NLCH3509.jpg --> 第6题

3. 应用题

4. 编程题

用html, css和js模拟实现一个下拉框，使得下拉框在各个浏览器下的样式和行为完全一致。说出你的设计方案，并且重点说吗功能设计时要考虑的因素。

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

<style type="text/css">

p,ul {

margin: 0;

}

ul {

padding: 0;

}

a {

text-decoration: none;

color: #000;

}

.btn {

border: 1px solid #000;

width: 100px;

text-align: center;

margin-bottom: 0.5px;

cursor: pointer;

padding: 2px 0;

}

.down {

width: 100px;

border: 1px solid #000;

text-align: center;

display: block;

}

.l\_sm {

border: 1px solid grey;

list-style: none;

padding: 2px 0;

cursor: pointer;

}

.l\_sm:hover {}

</style>

</head>

<body>

<p class="btn" id="btn">点击</p>

<ul class="down" id="show" >

<li class="l\_sm"><a href="#">第一项</a></li>

<li class="l\_sm"><a href="#">第二项</a></li>

<li class="l\_sm"><a href="#">第三项</a></li>

<li class="l\_sm"><a href="#">第四项</a></li>

</ul>

<script type="text/javascript">

var btn = document.getElementById("btn");

var show = document.getElementById("show");

btn.addEventListener("click", function() {

var style = window.getComputedStyle(show).display;

if(style=="block"){

show.style.display = "none";

} else {

show.style.display = "block";

}

});

</script>

</body>

</html>

4. 写出下面程序的运行结果（32位机器）：

void func(char \*str) {

printf("%d %d\n", sizeof(str), strlen(str));

}

main() {

char a[]="123456789";

printf("%d ", sizeof(a));

func(a);

}

5. 指定一个目录中，怎么查找文件名包含特定字符串的文件？

（用命令行或者其他方法）

6. 指定一个目录中，怎么查找内容包含特定字符串的文件？

（用命令行或者其他方法）

7. 在浏览器中输入www.baidu.com，从输入到页码显示出来，期间发生了哪些过程，详细描述一下（口述）

8. 谈一下你对设计模式的认识和理解，并简述几个你了解的设计模式（口述）

UPXT9452.jpg

1. 如何判断一个python对象的类型？

2. Python里面如何生成随机数？

3. 请写出匹配ip的python正则表达式

4. 写尽可能多的str对象的方法

5. 全局变量与局部变量的区别，如何中function里面给一个全局变量赋值？

6. tuple和list区别，有两个list，b1=[1,2,3]，b2=[2,3,4]，写出合并代码

7. 请写出一段python代码实现删除一个list里面的重复元素。l = [1,1,2,3,4,5,4]

8. 写出如下list的交集与差集代码，b1=[1,2,3]，b2=[2,3,4]

9. 请写出一段python代码实现list里排序a=[1,2,4,2,4,5,7,10,5,5,7,8,9,0,3]

UTSI4088.jpg

1. 简述TCP断开握手的过程

2. 简述 进程、线程、协程的含义。

3. 简述静态方法、类方法、实现方法的区别？

4. dict的items()方法与iteritems()方法的不同。

5. 请简述魔法函数\_\_new\_\_、\_\_call\_\_、\_\_rper\_\_、\_\_str\_\_、\_\_setattr\_\_、\_\_getattr\_\_的含义是什么？

6. django中如何读取和保存session，整个session的运行机制是什么？

VFXA6549.jpg

以下的题目如果是编程题，则可以任选你擅长的语言即可，也可以直接写伪代码。

1. 进程和线程的区别是什么？

2. A1, A2, ..., An 为n个整数，编写一个程序求出其最大值。

3. 找到一个目录下（包含子目录），所有的以”盈码基金“开头的，文件名包含

今天的日期的文件（日期格式为yyyy-mm-dd）。注意：文件名的编码均为utf-8。

4. 有两个字符串s1和s2，我们称对字符串的一次添加/删除/修改一个字符为一次操作。

如果我们可以采用最少的操作次数，把s1变成s2，这个操作次数称为s1和s2

之间的编辑距离，表示为ed(s1, s2)。比如s1="abc"，s2="bcd"，ed(s1, s2)=2。

请编写一个程序求出两个字符串之间的编辑距离。

VVCU3059.jpg

13. 根据当前栏目ID，生成面包屑。

栏目表结构：

CREATE TABLE `cms\_column` (

`column\_id` int(11) NOT NULL AUTO\_INCREMENT COMMENT '栏目ID',

`column\_name` varchar(45) DEFAULT NULL COMMENT '栏目名称',

`column\_path` varchar(255) NOT NULL DEFAULT '/' COMMENT '栏目目录',

`parent\_id` int(11) DEFAULT 'O' COMMENT '父ID',

`url` varchar(255) NOT NULL COMMENT `栏目URL',

PRIMARY KEY (`column\_id`)

)

要求输出面包屑格式： 首页 > 一级栏目 > 二级栏目 > ... > 当前栏目

14. 写出获得学生成绩报表的sql语句，涉及的数据库表如下：

。班级表（表名：classes）：

class\_id class\_name

----------------------

1 1班

2 2班

3 4班

。课程表（表名：lesson）：

lesson\_id lesson\_name

----------------------

1 语文

2 数学

3 化学

4 物理

。学生表（表名：student）：

student\_id student\_name class\_id

------------------------------------

1 张三 1

2 李四 2

3 王强 1

4 李明 3

。成绩表（表名：scores）：

score\_id student\_id lesson\_id score

---------------------------------------------

1 1 1 87

2 1 2 67

3 2 1 56

4 2 2 90

5 3 2 88

结果要求：

XOHT9171.jpg

11. MySQL中 delete from 表名 和truncate table 表名 的区别

12. 在MySQL中，要生成下列结果，SQL语句如何实现？

表名：com\_result

表内容：

rq shengfu

-------------------------

2017-04-09 胜

2017-04-09 胜

2017-04-09 负

2017-04-09 负

2017-04-10 胜

2017-04-10 负

2017-04-10 负

生成结果：

rq 胜 负

--------------------------

2017-04-09 2 2

2017-04-10 1 2

YJXH6200.jpg

1. 列表(list)和元组(tuple)的区别

2. 集合(set)是怎样的一种数据类型，何时使用？

3. 请在下面的空白处填写运行结果

>>> seq = [1,2,3,4]

>>> seq[:2]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>>> seq[-2:]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>>> seq[10:]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>>> seq[::-1]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>>> seq[:]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>>> id(seq[:]) == id(seq)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. 优化以下程序

result = []

for x in range(10):

result.append(x\*\*2)

print(result)

5. 函数、类方法定义中如何实现可选参数、可选关键词参数

6. 请解释classmethod和staticmethod的区别

7. 请给出下列程序的运行结果

>>> x=0.5

>>> whild x!=1.0:

... print(x)

... x+=0.1

8. 请写一段程序，包含迭代器和生成器的使用

9. 请根据age分别对 student\_tuples 及 student\_objects进行排序

>>> student\_tuples = [