Python写题经验

数据结构

常见数据结构时间复杂度

list

操作	案例	类型	备注
Index	l[i]	O(1)	
Store	l[i] = 0	O(1)	
Length	len(l)	O(1)	
Append	l.append(5)	O(1)	
Pop	l.pop()	O(1)	l.pop(-1) 同理
Clear	l.clear()	O(1)	l = [] 同理
Slice	l[a:b]	O(b-a)	
Extend	l.extend(...)	O(len(...))	取决于...的长度
Construction	list(...)	O(len(...))	取决于...的长度
Insert	l.insert(1)	O(N)	
Pop	l.pop(i)	O(N)	
check ==, !=	l1 == l2	O(N)	
Insert	l[a:b] = ...	O(N)	
Delete	del l[i]	O(N)	取决于i的位置，最坏的情况为O(N)
Containment	x in/not in l	O(N)	需逐个查找导致
Copy	l.copy()	O(N)	l[:] 同理
Remove	l.remove(...)	O(N)	
Extreme value	min(l)/max(l)	O(N)	需逐个查找导致
Reverse	l.reverse()	O(N)	
Iteration	for v in l:	O(N)	当循环中没有return或break的时候，为O(N)
Sort	l.sort()	O(N Log N)	
Multiply	k*l	O(k N)	

set

操作	案例	类型	备注
Length	len(s)	O(1)	
Add	s.add(5)	O(1)	
Containment	x in/not in s	O(1)	对比 list/tuple 的O(N) 快很多
Remove	s.remove(..)	O(1)	同上
Discard	s.discard(..)	O(1)	
Pop	s.pop()	O(1)	
Clear	s.clear()	O(1)	与 s = set() 同理
Construction	set(...)	O(len(...))	取决于 ... 的长度
check ==, !=	s != t	O(len(s))	与len(t)同理
<=/<	s <= t	O(len(s))	类似issubset方法
>=/>	s >= t	O(len(t))	
Union	s | t	O(len(s)+len(t))	
Intersection	s & t	O(len(s)+len(t))	
Difference	s - t	O(len(s)+len(t))	
Symmetric Diff	s ^ t	O(len(s)+len(t))	
Iteration	for v in s:	O(N)	当循环中没有return或break的时候，为O(N)
Copy	s.copy()	O(N)	

dict

操作	案例	类型	备注
Index	d[k]	O(1)	
Store	d[k] = v	O(1)	
Length	len(d)	O(1)	
Delete	del d[k]	O(1)	
get/setdefault	d.get(k)	O(1)	
Pop	d.pop(k)	O(1)	
Pop item	d.popitem()	O(1)	
Clear	d.clear()	O(1)	与 s = {} 或 s = dict() 同理
View	d.keys()	O(1)	与 d.values() 同理
Construction	dict(...)	O(len(...))	
Iteration	for k in d:	O(N)	包含以下方法: keys, values, items

deque

操作	案例	类型	备注
Copy	dq.copy()	O(n)	
append	dq.append(5)	O(1)	
appendleft	dq.appendleft(1)	O(1)	作用相当于 list.insert(0,1), 但后者效率为O(N)
pop	dq.pop()	O(1)	
popleft	dq.popleft()	O(1)	
extend	dq.extend(k)	O(len(k))	
extendleft	dq.extendleft(k)	O(len(k))	
rotate	dq.rotate(k)	O(len(k))	
remove	dq.remove(1)	O(n)	

queue

用collections库里面的deque模拟

https://blog.csdn.net/weixin_43790276/article/details/107749745

stack

用list模拟

堆

用heapq库封装。

```
import heapq


class PriorityQueue:
    def __init__(self):
        self.__queue = []

    def push(self, priority):
        heapq.heappush(self.__queue, priority)
        #这里可以通过正负号的设置重定义大小根堆

    def pop(self):
        return heapq.heappop(self.__queue)

    def front(self):
        return self.__queue[0]
```

输入输出

处理读入数组

```
n = int(input())
arr = [0] + list(map(int, input().split()))
temp = [0] * (n + 1)
```

快读快写

```
import sys
sys.stdin.readline()
sys.stdout.write()
```

数组初始化

```
N = int(1e5 + 10)
a = [0 for _ in range(N)]
#二维
s = [[0 for _ in range(N)] for _ in range(N)]
for i in range(n):
    arr.append(list(map(int, input().split())))
```

内置函数

bisect

```
from bisect import bisect_left, bisect_right

a = [0, 0, 1, 2, 2, 3]
```

```
# bisect_left返回如果将0插入a中，且维持数组顺序不变的条件下
#     放在最左边的索引。说人话就是a中第一个>=target的数的索引
index_1 = bisect_left(a, 0)

# bisect_right说人话就是a中最后一个>=target的数的索引+1
#
index_2 = bisect_right(a, 0)  # 注意返回的是实际的加1!


# 注意，可以添加查找范围，[start,end),左闭右开，不要将a变成a[start:end]引进去！拷贝会浪费大量的时间！！！
index_1 = bisect_left(a, 0, start,end)
index_2 = bisect_right(a, 0,start, end)
```

ascii()，chr()，ord()

与 ascii() 相关的，Python 还有两个内置函数：

- ord©：给定一个表示一个 Unicode 字符的字符串，返回一个表示该字符的 Unicode 码点的整数。例如 ord('a') 返回整数 97， ord('€') （欧元符号）返回 8364 。
- chr(i)：返回表示 Unicode 码位为整数 i 的字符的字符串。例如，chr(97) 返回字符串 'a'，chr(8364) 返回字符串 '€'。这是 ord() 的逆函数。实参的合法范围是 0 到 1,114,111（16 进制表示是 0x10FFFF）。如果 i 超过这个范围，会触发 ValueError 异常。

range() 函数的用法

```
range(stop)
range(start, stop[, step])
```

- Python3 range() 函数返回的是一个可迭代对象（类型是对象），而不是列表类型， 所以打印的时候不会打印列表。
- Python3 list() 函数是对象迭代器，可以把 range() 返回的可迭代对象转为一个列表，返回的变量类型为列表。

```
for number in range(1, 6):
    print(number,end=" ")
for number in range(1, 6, 2):
    print(number,end=" ")
for number in range(6):
    print(number,end=" ")
for number in range(6, 1, -1): %若步长为负数，开始值必须大于结束值
    print(number)
"""
1 2 3 4 5
1 3 5
0 1 2 3 4 5
6 5 4 3 2
"""
```

range()生成列表和元组

```
numbers1 = list(range(1, 6))
numbers2 = tuple(range(1, 6))
print(numbers1)
print(numbers2)
"""
[1, 2, 3, 4, 5]
(1, 2, 3, 4, 5)
"""
```

sort和sorted

sort 与 sorted 区别：

- sort 是应用在 list 上的方法，sorted 可以对所有可迭代的对象进行排序操作。
- list 的 sort 方法返回的是对已经存在的列表进行操作，而内建函数 sorted 方法返回的是一个新的 list，而不是在原来的基础上进行的操作。

sorted 语法：

```
sorted(iterable, key=None, reverse=False)  #默认升序reverse = false
```

先按照成绩降序排序，相同成绩的按照名字升序排序

```
d1 = [{'name':'alice', 'score':38}, {'name':'bob', 'score':18}, {'name':'darl',
'score':28}, {'name':'christ', 'score':28}]
l = sorted(d1, key=lambda x:(-x['score'], x['name']))
print(l)
```

sort()方法语法：

- 无返回值

```
list.sort( key=None, reverse=False)
```

map() 映射

- map() 函数会根据提供的函数对指定序列做映射。
- 第一个参数 function 以参数序列中的每一个元素调用 function 函数，返回包含每次 function 函数返回值的新列表。
- 返回一个迭代器。

`map(function, iterable, ...)`

- function -- 函数
- iterable -- 一个或多个序列

```
>>> def square(x) :          # 计算平方数
...     return x ** 2
...
>>> map(square, [1,2,3,4,5])    # 计算列表各个元素的平方
<map object at 0x100d3d550>     # 返回迭代器
>>> list(map(square, [1,2,3,4,5]))   # 使用 list() 转换为列表
[1, 4, 9, 16, 25]
>>> list(map(lambda x: x ** 2, [1, 2, 3, 4, 5]))   # 使用 lambda 匿名函数
[1, 4, 9, 16, 25]
>>>
```

len()返回长度或项目个数

- Python len() 方法返回对象（字符、列表、元组等）长度或项目个数。

```
>>>str = "runoob"
>>> len(str)             # 字符串长度
6
>>> l = [1,2,3,4,5]
>>> len(l)               # 列表元素个数
5
```

min()max()index()

- min() 方法返回给定参数的最小值，参数可以为序列
- 可返回区间最小值 (线段树效率更高吗？？)
- index() 函数用于从列表中找出某个值第一个匹配项的索引位置。
- **index**(*x*[, *start*[, *stop*]])（时间复杂度应该是O(n)）

返回最小的 *i* 使得 *i* 为数组中首次出现的 *x* 的索引号。 指定可选参数 *start* 和 *stop* 以便在数组的一个子部分内部搜索 *x*。

```
str = "runoob"
print ("最小字符: " + min(str))
a=[6,5,4,3,2,1]
i = a.index(min(a[1:4]),1,4)
ans = min(a[1:4])
print(i)
print(ans)
'''
最小字符: b
3
3
'''
```

翻转序列

```
[::-1]
```

字符串

```
s.lower()
s.upper()
s.isalpha()
s.isdigit()
```

全排列

```
from itertools import permutations

permutations(iterable, r=None)
```

参数说明： - `iterable`: 必需，表示要生成排列的序列或集合。 `list,dict,string`- `r`: 可选，表示每个排列的长度。如果不指定，则生成包含所有元素的排列。

日期处理

日期处理

```
from datetime import datetime,timedelta
#这里面常用的就是求两个日期中间隔了多少天
l = datetime(2000,1,1)
r = datetime(2020,10,1)
r - l # 默认的是显示天数，可以通过timedelta进行修改

#以及两个日期中间有多少个星期几什么的
#直接遍历
delta = timedelta(days=1)
while l != r:
    l += delta
```

字典计数器

https://blog.csdn.net/Judy18/article/details/124048121

math

部分函数	说明
math.pi	数学常数π= 3.141592...，以达到可用的精度
math.fabs(x)	返回x的绝对值。
math.factorial(x)	返回x阶乘。ValueError如果x不是整数或为负数则引发
math.gcd(a,b)	返回整数a和b的最大公约数
math.exp(x)	返回e**x
math.log(x)	返回x的自然对数（以e为底）
math.log10(x)	返回x的以10为底的对数
math.sqrt(x)	返回x的平方根
math.pow(x,y)	返回x**y，结果是实数
math.sin(x)	返回x弧度的正弦值