可借鉴答案

一些算法的规定写法也在答案里，复习时自取

考试有可能会考self practice和lab原题，复习时好好看

## Week 1

import time

start = time.clock()

import timeit

s1 = timeit.default\_timer()

e1 = timeit.default\_timer()

t1 = e1 - s1

True

python没有++，println

## Week 2

Import math

math.factorial(n)

math.comb(n, k)

str.lower()

str[::-1] 字符串反转

len(str)

list.pop()默认删最后一个 append默认往最后加 remove(element)

## Week 3

range 和 [0 : a] 这种都是前包后不包

mid = len(list) // 2 取整

min()

max()

list.count(element)

list.clear()

sorted(list) 不改原值，返回排序列表

list.sort() 改原值，不返回

list(str) string转list

"".join(list) list转string

dic.values() dic.keys() dict.items() 字典

if i in dic:

array[j], array[j-1] = array[j-1], array[j] 交换的简单写法

## Week 4

list.insert(0,new)

list.copy()

## Week 5

判断是不是None的时候:

if a is None: A为None，则if成立

if a == None: A为None，则if成立

if a: A为None，则if不成立

list(range(1,n+1)) 从1到n的列表

\_,c = linear(arr, i)

这个function返回的第一个值不要了，只关心第二个值为c

## Week 6

构造列表 / 矩阵：

self.hash\_table = [(None,None) for i in range(self.capacity)]

[[0 for j in range(column number)] for k in range(row number)]

类如果没有init方法，那么相当于只能创造一个object，就算你创造了多个，里面信息也是共通的

但是如果有init方法，只要管理好object的名字即可，不同名字就是不同object

Tuple也可以用[] 取值

a = (3,4)

a[0] = 3

\*矩阵乘法的写法

测空间占用：

import sys

sys.getsizeof(table)

import numpy as np

np.dot(matrix, matrix2) 矩阵乘法

np.add(matrix, matrix3) 矩阵加法

答案里有读取复杂file的方法

## Week 7

if not xxx:

self.id = list(range(0, 5))

self.color = [-1] \* 5

self.marked = [False for v in range(0, 5)]

dictionary.keys()

list[::-1] 输出reverse of list

import itertools

itertools.combinations(list, 2)

if v not in dict:

虽然只写了dict，但意思是dict.keys()也就是判断v在不在字典的keys里

dict.items()

for a,d in dict.items():

## Coursework:

print("hello", "world") hello world

print("hello" + "world") helloworld

## Week 8:

Linearithmic Nlog(N)

list.insert(index, element)

## Week 9:

Self里提到了Knuth-Morris-Pratt algorithm 和suffix trees 复习的时候看一下

String [] 的用法和list完全一样

s[index:index+length] 前包后不包

txt2 = "A" \* len(txt1) + "B"

index = ord(contact[level]) - ord('a')

chr(97 + index)

with open('5-mobydick.txt','r') as f:

txt1 = f.read().rstrip("\n")

from random import choice

from string import digits

str1 = "".join(choice(digits) for \_ in range(N[i]))

## Week 10

if c in string.punctuation or c in string.digits:

document = document.replace(c, "")

String.split() 根据空格，把string分成一个一个word的list

heapq.heappush() .heapreplace() .heappop()

result = sorted(result, key=lambda coor: coor[1])

Result是一个装元组的list，按照元组中第二个元素数值大小排序（从小到大）