作业3

栈类定义

class Stack:

def \_\_init\_\_(self):

self.list=[]

def IsEmpty(self):

if len(self.list)==0:

return True

else:

return False

def Push(self,x):

self.list.append(x)

def Pop(self):

if not self.IsEmpty():

return self.list.pop(len(self.list)-1)

else:

return None

def Top(self):

if not self.IsEmpty():

return self.list[len(self.list)-1]

else:

return None

def printstack(self):

print self.list

第一题

def testlist(L,A): #T1 #spcace complexity is 1; time comlexity is N+M

S=Stack()

i,j=0,0

while(True) :

if A[j] > S.Top() or S.Top() is None :

S.Push(L[i])

i+=1

#continue

elif A[j] == S.Top():

j+=1

S.Pop()

elif A[j] < S.Top():

return False

if j==len(A):

return True

L=[0,1,2,3,4,5,6,7,8,9] #出栈顺序

a=[4,3,2,1,0,9,8,7,6,5] #True

b=[4,6,8,7,5,3,2,9,0,1] #False

c=[2,5,6,7,4,8,9,3,1,0] #True

d=[4,3,2,1,0,5,6,7,8,9] #True

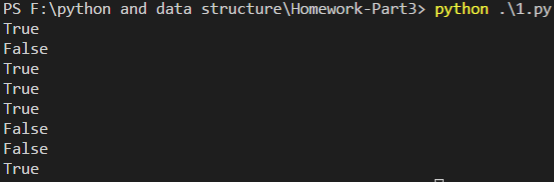
e=[1,2,3,4,5,6,9,8,7,0] #True

f=[0,4,6,5,3,8,1,7,2,9] #False

g=[1,4,7,9,8,6,5,3,0,2] #False

h=[2,1,4,3,6,5,8,7,9,0] #True

结果：



第二题

def testlist2(L,A): #T2

S=Stack()

i,j=0,0

while j<len(A):

if A[j] != S.Top() or S.Top() is None:

if i==len(L):

break

S.Push(L[i])

i+=1

elif A[j]==S.Top():

S.Pop()

j+=1

if S.IsEmpty() is True:

return True

else:

return False

L=['A','B','C','D','E']

A=['C','A','D','B']

print testlist2(L,A)

结果



第三题

def changestack(S1,S2): #T3

#l=S1.Len()

tmp=S1.Pop()

while True:

if tmp>S2.Top() or S2.IsEmpty():

S2.Push(tmp)

if S1.IsEmpty():break

tmp=S1.Pop()

elif tmp < S2.Top():

S1.Push(S2.Pop())

S2.printstack()

S1=Stack()

S2=Stack()

L=[2,5,6,7,4,8,9,3,1,0]

for a in L:

S1.Push(a)

changestack(S1,S2)

结果：



第四题

def changestack2(S): #T4

if S.IsEmpty():

return

top1=S.Pop()

if not S.IsEmpty():

changestack2(S)

top2 = S.Top()

if top1 < top2:

S.Pop()

S.Push(top1)

S.Push(top2)

changestack2(S)

return S

S.Push(top1)

S1=Stack()

S2=Stack()

L=[2,5,6,7,4,8,9,3,1,0]

for a in L:

S1.Push(a)

S2=changestack2(S1)

S2.printstack()

结果：



第五题

def changeorder(S): #T5

if S.IsEmpty():

return

top1 = S.Pop()

if not S.IsEmpty():

changeorder(S)

top2=S.Pop()

changeorder(S)

S.Push(top1)

changeorder(S)

S.Push(top2)

return S

S.Push(top1)

S1=Stack()

S2=Stack()

L=[2,5,6,7,4,8,9,3,1,0]

for a in L:

S1.Push(a)

S2=changeorder(S1)

S2.printstack()

结果：



第六题

def manacher(s): #时间复杂度为O(n)

if len(s)<=1:

return 0

t='?'

T=[t]

for c in s:

T.append(c)

T.append(t)

c,r,size = 1,2,len(T)

P=[0,1]+[None]\*(size-2)

maxIndex,maxCount=0,1

for i in range(2,size):

m=c\*2-i

if r>i and P[m]<r-i:

P[i]=P[m]

continue

count=min(i,size-i-1)

for n in range((1 if r <= i else r+1-i),count+1):

if T[i+n]!=T[i-n]:

count=n-1

break

c=i

r=i+count

P[i]=count

if count >maxCount:

maxCount=count

maxIndex=i-count

maxIndex=maxIndex //2

return maxCount

s='google'

print manacher(s)

结果如下

