SC：

1.A 2.C 3.C 4.D 5.D 6.A 7.D 8.C 9.A 10.C 11.A 12.A 13.D 14.D 15.B

MC:

1. B 17. A 18. AC 19.ABC 20.ACD 21.ABD 22.B 23.C 24.C 25.ABC

3.1)A.[10,7,6,5,4,3,2,1,0] 2points

B.O(n2) 2points

3.2)A.6 2points

B.1 2points

2

3

4

5

120

3.3) def power(x,n): recursion：4 pionts

partial=power(x,n//2)

result=partial\*partial O(logn):3 points

if n%2==1:

result=result\*x

return result

3.4) class MyStack(object):

def \_\_init\_\_(self,max): 1 points

self.head=-1

self.stack=list()

self.max=max

for i in range(self.max):

self.stack.append(0)

def put(self,item): 2 points

if self.head>=self.max:

return 'Put Error: The Stack is Overflow!'

else:

self.head+=1

self.stack[self.head]=item

print ('Put %s Success' %item)

def get(self): 2 points

if self.head<0:

return 'Get Error: The Stack is Empty!'

else:

self.head-=1

return self.stack[self.head+1]

def isEmpty(self): 2 points

if self.head<-1:

return True

return False

3.5)

A: Depth First Search 2points

B: O(n) 2points

C: multiple recursive function 2points

D: 8 2points

10

14

13

3

6

7

4

1