以下题目中的tf代表python中的tensorflow包。

1. Tensor A的shape为（2,6），A的rank是多少。
2. Tensor A为[[1,2],[4,5]]，Tensor B为[[2,1],[3,2]]，tf.matmul(A,B)返回的Tensor应该是什么。
3. 仍然是题目2中的两个Tensor，tf.multiply(A,B)返回的Tensor是什么。
4. 对于一个Tensor A，tf.square(A)返回的是A的什么运算。
5. A的shape为（2,3,4,5），B的shape为（5,4,3,2），A和B可以进行tf.matmul(A,B)运算吗。如果可以，tf.matmul(A,B)返回的Tensor的shape是什么。（Tensor的矩阵运算）
6. A的shape为（2,3,4,5），B的shape为（2,3,5,4），A和B可以进行tf.matmul(A,B)运算吗。如果可以，tf.matmul(A,B)返回的Tensor的shape是什么。（Tensor的矩阵运算）
7. A的shape为（4,5），B的shape为（5，），A和B可以进行tf.multiply(A,B)运算吗。如果可以，tf. multiply (A,B)返回的Tensor的shape是什么。(broadcasting mechanism)
8. A的shape为（2,1,4,5），B的shape为（2,3,4,1），A和B可以进行tf.multiply(A,B)运算吗。如果可以，tf. multiply (A,B)返回的Tensor的shape是什么。(broadcasting mechanism)
9. Tensorflow的session运行时，默认占用所有可见GPU 100%的显存（减少显存碎片，提升运行速度），如何配置tensorflow的session才能使其动态分配内存，而不是全部占用？如何配置可以给session固定分配40%的显存？
10. Tensor A和B具有相同的shape，求一个Tensor C，C的shape和A、B相同，C中的元素是A、B中对应元素较大的那一个。例如A=[[1,5],[7,10]]，B=[[2,3],[8,9]]，则C=[[2,5],[8,10]]。请构造一个计算图去计算C。
11. A为TensorFlow中的一个placeholder，其shape为[2,3,None]（即最后一个维度的大小根据传入(feed)的A的Tensor而定，是动态变化的）。如何在tensorflow的计算图中动态的根据A的第2维(维度从0开始，第0维、第1维…)的大小遍历Tensor。
12. A为TensorFlow中的一个placeholder，其shape为[2,3,None]（即最后一个维度的大小根据传入(feed)的A的Tensor而定，是动态变化的）。如何在tensorflow的计算图中动态的判断A的第2维(维度从0开始，第0维、第1维…)的大小是否为25。

**def** tf\_scope\_test():

with tf.name\_scope('namescope'):

var\_1 = tf.Variable(initial\_value=[0], name='var\_1')

var\_2 = tf.get\_variable(name='var\_2', shape=[1, ])

with tf.variable\_scope('variable\_scope'):

var\_3 = tf.Variable(initial\_value=[0], name='var\_3')

var\_4 = tf.get\_variable(name='var\_4', shape=[1, ])

print(var\_1.name)

print(var\_2.name)

print(var\_3.name)

print(var\_4.name)

这个函数的输出应该是什么？



**def** tf\_add\_node\_mem\_test(): *# 新建op节点*

ini = tf.constant\_initializer(1, dtype=tf.int32)

w = tf.get\_variable("test", [1],

initializer=ini,

dtype=tf.int32)

i = 0

with tf.Session() as sess:

sess.run(tf.global\_variables\_initializer())

x = tf.placeholder(tf.int32)

while True:

op = tf.multiply(w, x)

tmp = sess.run(op, feed\_dict={x: i})

i += 1

if i % 200 == 0:

print(tmp)

**def** tf\_reuse\_node\_mem\_test(): *# 复用节点*

ini = tf.constant\_initializer(1, dtype=tf.int32)

w = tf.get\_variable("test", [1],

initializer=ini,

dtype=tf.int32)

i = 0

with tf.Session() as sess:

sess.run(tf.global\_variables\_initializer())

x = tf.placeholder(tf.int32)

op = tf.multiply(w, x)

while True:

tmp = sess.run(op, feed\_dict={x: i})

i += 1

if i % 200 == 0:

print(tmp)

上面两个函数，哪一个的时间效率高，哪一个空间效率高，为什么

1. A=[[1,2,3],[1,2,3]]，tf.reduce\_mean(A)的结果是什么。
2. A=[[1,2,3],[1,2,3]]，tf.reduce\_mean(A,axis=0)的结果是什么。
3. A=[[1,2,3],[1,2,3]]，tf.reduce\_mean(A,axis=1)的结果是什么。
4. A=[[1,2,3],[1,2,3]]，tf.reduce\_mean(A,axis=0,keepdims=True)的结果是什么。
5. A=[[1,2,3],[1,2,3]]，tf.reduce\_mean(A,axis=[0,1])的结果是什么。
6. A的shape为(2,3,4)，元素全为1，tf.reduce\_mean(A,axis=0)结果的shape是什么。
7. A的shape为(2,3,4)，元素全为1，tf.reduce\_mean(A,axis=1) 结果的shape是什么。
8. A的shape为(2,3,4)，元素全为1，tf.reduce\_mean(A,axis=2) 结果的shape是什么。
9. A的shape为(2,3,4)，元素全为1，tf.reduce\_mean(A,axis=[0,1]) 结果的shape是什么。
10. A的shape为(2,3,4)，元素全为1，tf.reduce\_mean(A,axis=2,keepdims=True) 结果的shape是什么。
11. A的shape为(2,3,4)，元素全为1，tf.reduce\_mean(A,axis=[0,1],keepdims=True) 结果的shape是什么。
12. A的shape为（5,7,8），tf.split(A,2,axis=-1)结果的shape是什么。
13. A的shape为（5,7,8），tf.split(A,2,axis=1)结果的shape是什么。
14. A的shape为（5,7,8），tf.split(A,[2,2,4],axis=-1)结果的shape是什么。
15. A的shape为（2,3），B的shape为（2,5），C=tf.concat([A,B],axis=-1)，C的shape是什么。
16. A的shape为（2,3），B的shape为（2,5），C=tf.concat([A,B],axis=0)，C的shape是什么。
17. 如何清空使用tf.add\_to\_collection(key,val)添加到Graph中的集合。