

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  $\begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$ ，Tensor B 为  $\begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix}$ ，`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  $A = \begin{bmatrix} 1 & 5 \\ 7 & 10 \end{bmatrix}$ ， $B = \begin{bmatrix} 2 & 3 \\ 8 & 9 \end{bmatrix}$ ，则  $C = \begin{bmatrix} 2 & 5 \\ 8 & 10 \end{bmatrix}$ 。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。

13.

```
def tf_scope_test():
    with tf.name_scope('namespace'):
        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？

14.

```
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

15.  $A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$ ,  $\text{tf.reduce\_mean}(A)$ 的结果是什么。
16.  $A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$ ,  $\text{tf.reduce\_mean}(A, \text{axis}=0)$ 的结果是什么。
17.  $A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$ ,  $\text{tf.reduce\_mean}(A, \text{axis}=1)$ 的结果是什么。
18.  $A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$ ,  $\text{tf.reduce\_mean}(A, \text{axis}=0, \text{keepdims}=\text{True})$ 的结果是什么。
19.  $A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$ ,  $\text{tf.reduce\_mean}(A, \text{axis}=[0,1])$ 的结果是什么。
20.  $A$  的 shape 为 (2,3,4), 元素全为 1,  $\text{tf.reduce\_mean}(A, \text{axis}=0)$  结果的 shape 是什么。
21.  $A$  的 shape 为 (2,3,4), 元素全为 1,  $\text{tf.reduce\_mean}(A, \text{axis}=1)$  结果的 shape 是什么。
22.  $A$  的 shape 为 (2,3,4), 元素全为 1,  $\text{tf.reduce\_mean}(A, \text{axis}=2)$  结果的 shape 是什么。
23.  $A$  的 shape 为 (2,3,4), 元素全为 1,  $\text{tf.reduce\_mean}(A, \text{axis}=[0,1])$  结果的 shape 是什么。
24.  $A$  的 shape 为 (2,3,4), 元素全为 1,  $\text{tf.reduce\_mean}(A, \text{axis}=2, \text{keepdims}=\text{True})$  结果的 shape 是什么。
25.  $A$  的 shape 为 (2,3,4), 元素全为 1,  $\text{tf.reduce\_mean}(A, \text{axis}=[0,1], \text{keepdims}=\text{True})$  结果的 shape 是什么。
26.  $A$  的 shape 为 (5,7,8),  $\text{tf.split}(A, 2, \text{axis}=-1)$  结果的 shape 是什么。
27.  $A$  的 shape 为 (5,7,8),  $\text{tf.split}(A, 2, \text{axis}=1)$  结果的 shape 是什么。
28.  $A$  的 shape 为 (5,7,8),  $\text{tf.split}(A, [2,2,4], \text{axis}=-1)$  结果的 shape 是什么。
29.  $A$  的 shape 为 (2,3),  $B$  的 shape 为 (2,5),  $C = \text{tf.concat}([A, B], \text{axis}=-1)$ ,  $C$  的 shape 是什么。
30.  $A$  的 shape 为 (2,3),  $B$  的 shape 为 (2,5),  $C = \text{tf.concat}([A, B], \text{axis}=0)$ ,  $C$  的 shape 是什么。
31. 如何清空使用  $\text{tf.add\_to\_collection}(\text{key}, \text{val})$  添加到 Graph 中的集合。