三对角矩阵是指除了主对角线及其上下两条对角线上有非零元素外，其余元素均为零的矩阵。因此，对于一个 \( n \times n \) 的三对角矩阵，只需要存储这三条对角线上的元素即可。

设三对角矩阵为 \( A \)，其元素为 \( a\_{ij} \)，其中 \( i, j \in [1, n] \)。现在我们要将其存储在一个三维数组 \( B \) 中，其中 \( B[i][j][k] \) 表示第 \( k \) 条对角线上的第 \( j \) 个元素，且 \( i \) 是行索引，这里 \( i \) 总是等于 0 或 1，因为我们只关心三条对角线。

### 步骤分析

1. \*\*确定对角线类型\*\*:

- 主对角线: 当 \( i = j \) 时，即 \( a\_{ii} \)。

- 上对角线: 当 \( i = j - 1 \) 时，即 \( a\_{(i+1)i} \)。

- 下对角线: 当 \( i = j + 1 \) 时，即 \( a\_{i(j+1)} \)。

2. \*\*映射到数组 B\*\*:

- 对于主对角线，\( B[0][i][i] = a\_{ii} \)。

- 对于上对角线，\( B[1][i][i-1] = a\_{(i+1)i} \)。

- 对于下对角线，\( B[2][i][i+1] = a\_{i(j+1)} \)。

### 编程实现

下面给出一个简单的 Python 实现，用于将三对角矩阵转换成指定格式的数组。

```python

def convert\_to\_B(A):

n = len(A)

B = [[[None for \_ in range(n)] for \_ in range(n)] for \_ in range(3)]

# 填充主对角线

for i in range(n):

B[0][i][i] = A[i][i]

# 填充上对角线

for i in range(1, n):

B[1][i][i-1] = A[i][i-1]

# 填充下对角线

for i in range(n-1):

B[2][i][i+1] = A[i][i+1]

return B

# 测试代码

A = [

[1, 2, 0, 0],

[3, 4, 5, 0],

[0, 6, 7, 8],

[0, 0, 9, 10]

]

B = convert\_to\_B(A)

print(B)

```

这段代码首先定义了一个函数 `convert\_to\_B`，它接受一个三对角矩阵 `A` 并返回一个三维数组 `B`。然后，我们创建了一个测试矩阵 `A` 并调用了这个函数，最后打印出结果。

注意，这里的实现假设输入矩阵 `A` 已经是一个标准的三对角矩阵，并且大小为 \( n \times n \)。如果输入不是这样的情况，则需要额外处理或者修改函数逻辑。