“返回一个包含输入 input 中非零元素索引的张量.输出张量中的每行包含 input 中非零元素的索引.
如果输入 input 有 n 维,则输出的索引张量 out 的 size 为 z x n , 这里 z 是输入张量 input 中所有非零元素的个数.”

其实连着例子看的话，第一句话是很好理解的，就是第二句话有点搞不清楚究竟想说什么。

那么看一下文档给的第二个例子，

```
>>> torch.nonzero(torch.Tensor([[0.6, 0.0, 0.0, 0.0],
...                             [0.0, 0.4, 0.0, 0.0],
...                             [0.0, 0.0, 1.2, 0.0],
...                             [0.0, 0.0, 0.0,-0.4]]))

 0  0
 1  1
 2  2
 3  3
[torch.LongTensor of size 4x2]
```

这里直接给出理解方式，输出的这个out应该这样去读：这个例子里input是2维的，一共有4个非0元素，所以输出是一个4×2的张量，表示每个非0元素的索引。读法是从左往右，比如out的第0行[0,0]，表示的就是input的第0行的第0个元素是非0元素，同理，out的第1行[1,1]，表示的就是input的第1行的第1个元素是非0元素，等等。

那比如我们再把input设为一个3维张量，举个例子：

```
torch.nonzero(torch.Tensor([[[1,1,1,0,1],[1,0,0,0,1]],
                            [[1,1,1,0,1],[1,0,0,0,1]]]))
```

输出的结果是：

```
   0    0    0
   0    0    1
   0    0    2
   0    0    4
   0    1    0
   0    1    4
   1    0    0
   1    0    1
   1    0    2
   1    0    4
```

```
   1    1    0
   1    1    4
[torch.LongTensor of size 12x3]
```

这个out张量的意思就是：

按行依次从左往右读，第0行第0列第0个元素非0，第0行第0列第1个元素非0，……，第1行第1列第0个元素非0，第1行第1列第4个元素非0.

这就是torch.nonzero的理解方式。