Tensors - Shape, Dimension, Rank

Tensors - Shape, Dimension, Rank

◈Shape, Rank, Dimension

```
a1 = torch.tensor(1)                                    # shape: torch.Size([]), ndims(=rank): 0

a2 = torch.tensor([1])                                  # shape: torch.Size([1]), ndims(=rank): 1

a3 = torch.tensor([1, 2, 3, 4, 5])                      # shape: torch.Size([5]), ndims(=rank): 1

a4 = torch.tensor([[1], [2], [3], [4], [5]])            # shape: torch.Size([5, 1]), ndims(=rank): 2

a5 = torch.tensor([                                     # shape: torch.Size([3, 2]), ndims(=rank): 2
    [1, 2],
    [3, 4],
    [5, 6]
])

a6 = torch.tensor([                                     # shape: torch.Size([3, 2, 1]), ndims(=rank): 3
    [[1], [2]],
    [[3], [4]],
    [[5], [6]]
])
```

Tensors - Shape, Dimension, Rank

◈Shape, Rank, Dimension

```
a7 = torch.tensor([                    # shape: torch.Size([3, 1, 2, 1]), ndims(=rank): 4
    [[[1], [2]]],
    [[[3], [4]]],
    [[[5], [6]]]
])

a8 = torch.tensor([                    # shape: torch.Size([3, 1, 2, 3]), ndims(=rank): 4
    [[[1, 2, 3], [2, 3, 4]]],
    [[[3, 1, 1], [4, 4, 5]]],
    [[[5, 6, 2], [6, 3, 1]]]
])


a9 = torch.tensor([                    # shape: torch.Size([3, 1, 2, 3, 1]), ndims(=rank): 5
    [[[[1], [2], [3]], [[2], [3], [4]]]],
    [[[[3], [1], [1]], [[4], [4], [5]]]],
    [[[[5], [6], [2]], [[6], [3], [1]]]]
])
```

Tensors - Shape, Dimension, Rank

◈Shape, Rank, Dimension

```
a10 = torch.tensor([                    # shape: torch.Size([4, 5]), ndims(=rank): 2
    [1, 2, 3, 4, 5],
    [1, 2, 3, 4, 5],
    [1, 2, 3, 4, 5],
    [1, 2, 3, 4, 5],
])

a10 = torch.tensor([                    # shape: torch.Size([4, 1, 5]), ndims(=rank): 3
    [[1, 2, 3, 4, 5]],
    [[1, 2, 3, 4, 5]],
    [[1, 2, 3, 4, 5]],
    [[1, 2, 3, 4, 5]],
])

a11 = torch.tensor([                    # ValueError: expected sequence of length 3 at dim 3 (got 2)
    [[1, 2, 3], [4, 5]]],
    [[1, 2, 3], [4, 5]]],
    [[1, 2, 3], [4, 5]]],
    [[1, 2, 3], [4, 5]]],
])
```