Numpy.linalg.\*

TLDR: implementing matmul() is enough for most applications

1. Dot (a, b)
   1. 1d-1d: inner() -> matmul()
   2. 2d-2d: matmul(): implement
   3. 0d-nd: no implement
   4. Nd-1d: no implement
   5. Nd-Md: no implement
2. Multi\_dot (a)
   1. Len(a) == 2: dot()
   2. Len(a) == 3: \_multi\_dot\_three() -> dot()
   3. Len(a) >3: recursive call
3. Vdot(a, b)
   1. Have complex numbers: no implement
   2. 1d-1d: dot()
   3. Nd-nd: flat() -> dot()
4. Inner(a, b)
   1. 1d-1d: matmul (a, transpose(b)) -> matmul()
   2. Others: no implement
5. Outer(a, b)
   1. 1d-1d: matmul (transpose(a), b) -> matmul()
   2. Others: no implement
6. Matmul(a, b)
   1. 2d-2d: implement
   2. 2d-1d: implement
   3. 3d-2d: implement in the future
   4. Others: no implement
7. Tensordot(a, b, axes)
   1. Dot() ??? implement
8. Einsum
   1. Implement in the future
9. Einsum\_path
   1. Implement in the future
10. Matrix\_power (a, n)
    1. Dot() implement