

Abubakkar Abdullah

Subject Name – Numerical Computing

Section-5A

RollNo-20p-0045


```
In [6]: class Vector:
        def __init__(self ,x = 0.0 ,y = 0.0):
            self.x = x
            self.y = y

        def __str__(self):
            return "[{} , {}]".format(str(self.x),str(self.y))
```

```
In [7]: a = Vector(2 , 4)
```

```
In [ ]: print(a)
```

```
In [ ]: b = Vector(5 ,2)
        print(b)
```

```
In [3]: def add(self ,b):
        c = Vector()
        c.x = self.x+b.x
        c.y = self.y+b.y
        return c
```

```
Vector.add=add
```

```
In [ ]: c = a.add(b)
        print(c)
```

```
In [4]: def mul(self ,s):
        return Vector(s * self.x ,s*self.y)
```

```
Vector.mul = mul
```

```
In [ ]: d = a.mul(2)
        print(d)
```

```
In [5]: def sub(self,b):|
        return self.add(b.mul(-1))
```

```
Vector.sub = sub
```

```
In [ ]: d_min_b=d.sub(b)
        print(d_min_b)
```

```
In [ ]: d_min b=d.sub(b)
        print(d_min_b)
```

```
In [ ]: class Matrix:
        def __init__(self,dims,fill):
            self.rows=dims[0]
            self.cols=dims[1]
            self.A=[
                [fill] * self.cols
                for i in range(self.rows)
            ]
```

```
In [ ]: m=Matrix((3,4),2.0)
```

```
In [ ]: print(m)
```

```
In [ ]: def __str__(self):
        rows=len(self.A)
        ret = ''

        for i in range(rows):
            cols=len(self.A[i])
            for j in range(cols):
                ret+=str(self.A[i][j])+"\t"
            ret+="\n"

        return ret

Matrix.__str__=__str__
```

```
In [ ]: print(m)
```

```
In [ ]: %time n = Matrix((100,100),0.0)
```

```
In [ ]: from sys import getsizeof
        print(getsizeof(m))
        print(getsizeof(n))
```

```
In [ ]: !pip install pympler
```

```
In [ ]: from pympler.asizeof import asizeof
```

```
In [ ]: asizeof(m),asizeof(n)
```

In [24]: `print(m)`

```
2.0  2.0  2.0  2.0
2.0  2.0  2.0  2.0
2.0  2.0  2.0  2.0
```

In [25]: `%time n = Matrix((100,100),0.0)`

```
CPU times: user 53 µs, sys: 30 µs, total: 83 µs
Wall time: 88.2 µs
```

In [26]: `from sys import getsizeof
print(getsizeof(m))
print(getsizeof(n))`

```
48
48
```

In [27]: `!pip install pympler`

```
Requirement already satisfied: pympler in /home/spoofy/.local/lib/python3.8/site-packages (1.0.1)
```

In [28]: `from pympler.asizeof import asizeof`

In [29]: `asizeof(m),asizeof(n)`

Out[29]: (760, 86880)

In [30]: `dim=5000`

In [31]: `%time m=Matrix((dim,dim),0.0)`

```
CPU times: user 205 ms, sys: 80.2 ms, total: 285 ms
Wall time: 283 ms
```

In [32]: `size=asizeof(m)/(1024*1024)
print("{:.2f} MBs.".format(size))`

```
191.04 MBs.
```

```
In [33]: def get(self,i,j):
        if i<0 or i>self.rows:
            raise ValueError("Row index out of range.")
        if j<0 or j>self.cols:
            raise ValueError("Column index out of range.")

        if(i,j) in self.vals:
            return self.vals[(i,j)]

        return 0.0

Matrix.get=get
```

```
In [34]: m.get(1,2)
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-34-6f98ca7ab6a9> in <module>
----> 1 m.get(1,2)

<ipython-input-33-b11b417bdd50> in get(self, i, j)
      5         raise ValueError("Column index out of range.")
      6
----> 7     if(i,j) in self.vals:
      8         return self.vals[(i,j)]
      9

AttributeError: 'Matrix' object has no attribute 'vals'
```

```
In [ ]: m.get(15,0)
```

```
In [ ]: m.get(1,10)
```

```
In [42]: m.set(1,2,15.0)
```

```
In [43]: m.get(1,2)
```

```
Out[43]: 15.0
```

```
In [44]: m.vals
```

```
Out[44]: {(1, 2): 15.0}
```

```
In [45]: m.set(1,4,29.9)
```

```
In [46]: m.get(1,4)
```

```
Out[46]: 29.9
```

```
In [47]: dim=500  
m=Matrix((dim,dim))
```

```
In [48]: asizeof(m)
```

```
Out[48]: 416
```

```
In [49]: %matplotlib inline  
%run mlimp.py
```

```
ERROR:root:File `mlimp.py` not found.
```

```
In [50]: import numpy as np
```

```
In [51]: np.random.seed(1337)
```

```
In [52]: x=np.array([1,4,3])
x
```

```
Out[52]: array([1, 4, 3])
```

```
In [53]: y=np.array([[1,4,3],
                    [9,2,7]])
y
```

```
Out[53]: array([[1, 4, 3],
               [9, 2, 7]])
```

```
In [54]: x.shape
```

```
Out[54]: (3,)
```

```
In [55]: y.shape
```

```
Out[55]: (2, 3)
```

```
In [56]: z=np.array([[1,4,3]])
```

```
In [57]: z.shape
```

```
Out[57]: (1, 3)
```

```
In [58]: z=np.arange(1,2000,1)
z[:-10]
```

```
Out[58]: array([ 1,  2,  3, ..., 1987, 1988, 1989])
```

```
In [59]: z.shape
```

```
Out[59]: (1999,)
```

```
In [60]: np.arange(0.5,3,0.5)
```

```
Out[60]: array([0.5, 1. , 1.5, 2. , 2.5])
```

```
In [61]: np.arange(0.5,10,1).shape
```

```
Out[61]: (10,)
```

```
In [ ]: np.arange(0.5,10,1).reshape(5,2).shape
```


Out[62]: (5, 2)

```
In [63]: np.arange(0.5,10,1).reshape(5,3).shape
```

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-63-84a01f6ea824> in <module>  
----> 1 np.arange(0.5,10,1).reshape(5,3).shape  
  
ValueError: cannot reshape array of size 10 into shape (5,3)
```

```
In [64]: np.linspace(3,9,10)
```

Out[64]: array([3. , 3.66666667, 4.33333333, 5. , 5.66666667,
6.33333333, 7. , 7.66666667, 8.33333333, 9.])

```
In [65]: print(x)  
print(x[1])  
print(x[1:])
```

```
[1 4 3]  
4  
[4 3]
```

```
In [66]: print(y)  
y[0,1]
```

```
[[1 4 3]  
 [9 2 7]]
```

Out[66]: 4

```
In [67]: print(y)  
y[0][1]
```

```
[[1 4 3]  
 [9 2 7]]
```

Out[67]: 4

```
In [68]: y[:,1]
```

Out[68]: array([4, 2])

```
In [69]: y[:,[1,2]]
```

```
In [64]: np.linspace(3,9,10)
```

```
Out[64]: array([3.          , 3.66666667, 4.33333333, 5.          , 5.66666667,  
               6.33333333, 7.          , 7.66666667, 8.33333333, 9.          ])
```

```
In [65]: print(x)  
         print(x[1])  
         print(x[1:])
```

```
[1 4 3]  
4  
[4 3]
```

```
In [66]: print(y)  
         y[0,1]
```

```
[[1 4 3]  
 [9 2 7]]
```

```
Out[66]: 4
```

```
In [67]: print(y)  
         y[0][1]
```

```
[[1 4 3]  
 [9 2 7]]
```

```
Out[67]: 4
```

```
In [68]: y[:,1]
```

```
Out[68]: array([4, 2])
```

```
In [69]: y[:,[1,2]]
```

```
Out[69]: array([[4, 3],  
               [2, 7]])
```


