Assignment 3

October 10, 2018

0.1 By - Abhey Arora, Roll Number - BS-1618

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
```

```
In [4]: jar = pd.read_csv("F:/abhey/B3/Algorithms/Assignment/Assignment 3/RunTimeJarvis.csv")
        jar
```

```
Out[4]:     n=5000  n=10000  n=20000  n=50000  n=70000  n=100000
        0   0.0060   0.0000  0.01600   0.0150   0.0160    0.0310
        1   0.0000   0.0000  0.00000   0.0160   0.0160    0.0220
        2   0.0000   0.0150  0.01600   0.0160   0.0310    0.0160
        3   0.0000   0.0000  0.00000   0.0150   0.0160    0.0160
        4   0.0000   0.0000  0.00000   0.0160   0.0220    0.0160
        5   0.0000   0.0000  0.01600   0.0150   0.0150    0.0220
        6   0.0000   0.0000  0.00000   0.0070   0.0160    0.0630
        7   0.0000   0.0000  0.00000   0.0160   0.0160    0.0530
        8   0.0000   0.0160  0.01500   0.0150   0.0160    0.0160
        9   0.0000   0.0000  0.00000   0.0160   0.0220    0.0370
        10  0.0000   0.0000  0.00000   0.0150   0.0160    0.0160
        11  0.0160   0.0000  0.01600   0.0160   0.0150    0.0150
        12  0.0000   0.0160  0.00000   0.0160   0.0160    0.0320
        13  0.0000   0.0000  0.00000   0.0060   0.0220    0.0530
        14  0.0000   0.0000  0.01500   0.0160   0.0330    0.0620
        15  0.0000   0.0000  0.00000   0.0150   0.0250    0.0230
        16  0.0000   0.0000  0.01600   0.0160   0.0160    0.0310
        17  0.0000   0.0150  0.00700   0.0160   0.0220    0.0310
        18  0.0000   0.0000  0.00000   0.0150   0.0320    0.0220
        19  0.0000   0.0000  0.00000   0.0160   0.0150    0.0310
        20  0.0011   0.0031  0.00585   0.0147   0.0199    0.0304
```

```
In [5]: gra = pd.read_csv("F:/abhey/B3/Algorithms/Assignment/Assignment 3/RunTimeGraham.csv")
        gra
```

```
Out[5]:      n=5000  n=10000  n=20000  n=50000  n=70000  n=100000
        0   0.00000   0.0160   0.0160  0.03800   0.0690   0.06900
        1   0.00000   0.0000   0.0060  0.01600   0.0470   0.04700
        2   0.00000   0.0000   0.0000  0.03100   0.0320   0.05300
```

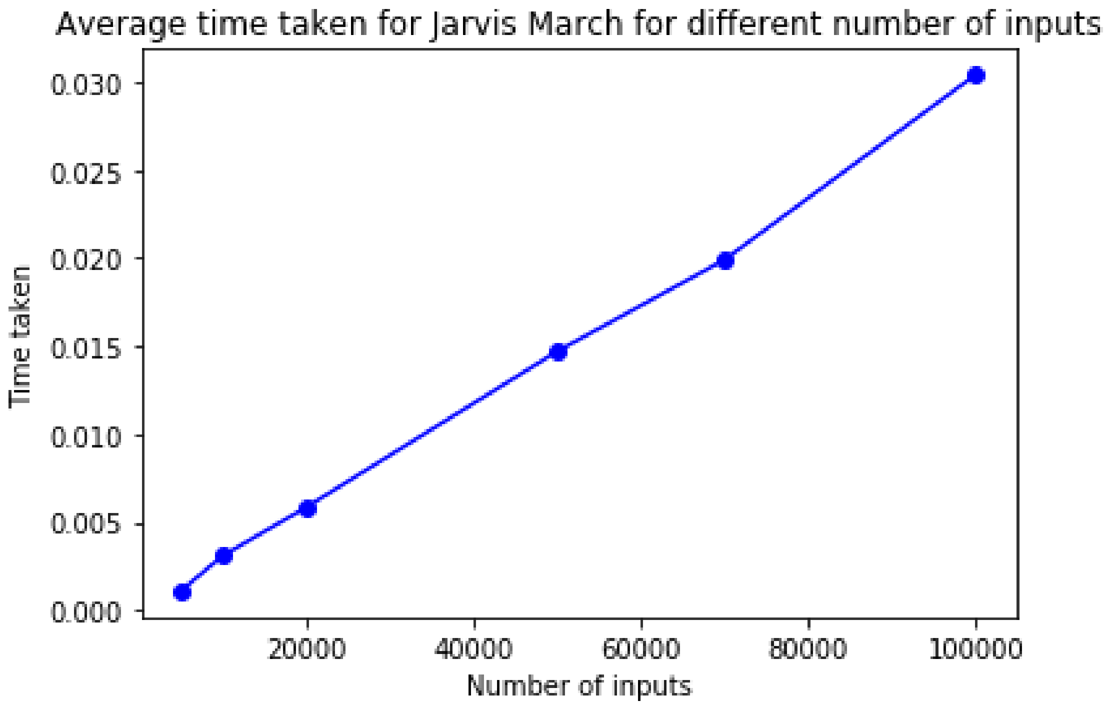
```
3   0.00000   0.0160   0.0160  0.01600   0.0530   0.04700
4   0.00000   0.0000   0.0160  0.03100   0.0310   0.03700
5   0.00000   0.0060   0.0000  0.02200   0.0540   0.04700
6   0.00000   0.0000   0.0150  0.01500   0.0620   0.03800
7   0.01600   0.0000   0.0000  0.04700   0.0380   0.09300
8   0.00000   0.0160   0.0000  0.04500   0.0690   0.06000
9   0.00000   0.0000   0.0160  0.05500   0.0780   0.10200
10  0.00000   0.0150   0.0000  0.03100   0.0590   0.05300
11  0.00000   0.0000   0.0150  0.03100   0.0390   0.06900
12  0.00000   0.0160   0.0000  0.02200   0.0520   0.08500
13  0.00000   0.0000   0.0150  0.04700   0.0460   0.06200
14  0.01500   0.0000   0.0000  0.03100   0.0540   0.10000
15  0.00000   0.0160   0.0160  0.02300   0.0310   0.05400
16  0.00000   0.0000   0.0160  0.03100   0.0380   0.06900
17  0.00000   0.0000   0.0000  0.01600   0.0620   0.10000
18  0.00000   0.0150   0.0150  0.02200   0.0850   0.09400
19  0.00000   0.0000   0.0000  0.03100   0.0850   0.03600
20  0.00155   0.0058   0.0081  0.03005   0.0542   0.06575
```

Both of the above DataFrames have the first 20 rows as the 20 iterations of a single n-value. After that, the 21^{st} row contains the average value for that particular n-value.

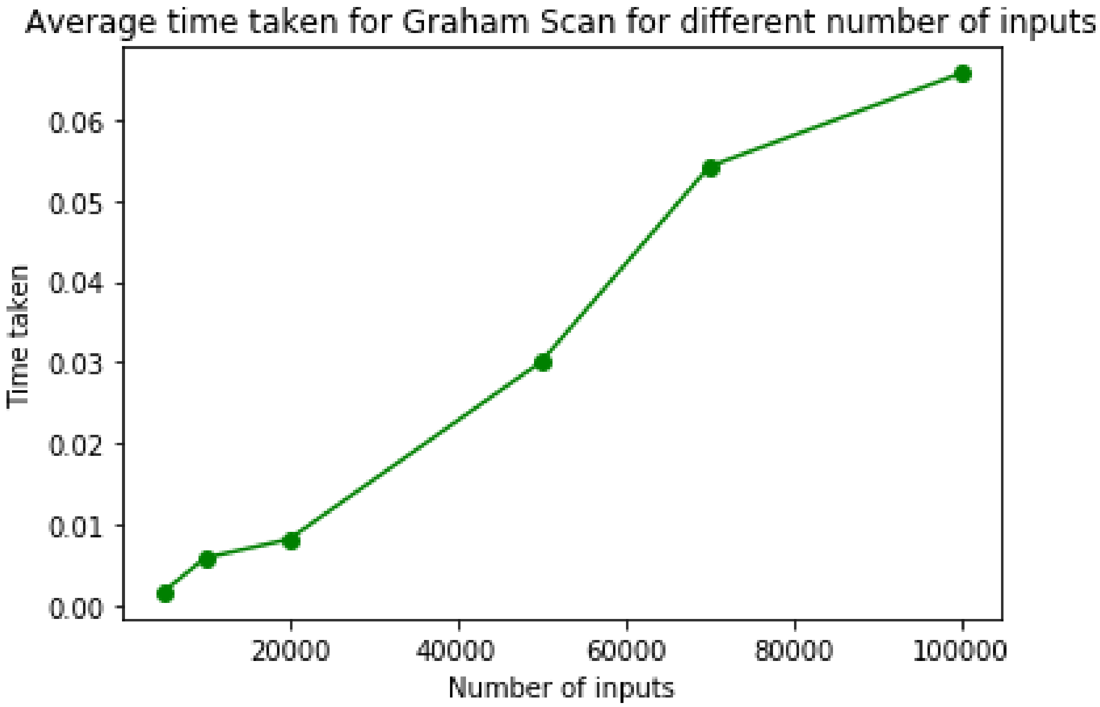
```
In [8]: n = [5000,10000,20000,50000,70000,100000]
        jaravg = jar.iloc[20]
        graavg = gra.iloc[20]
```

Now, we will plot the average time taken by each algorithm against number of inputs taken

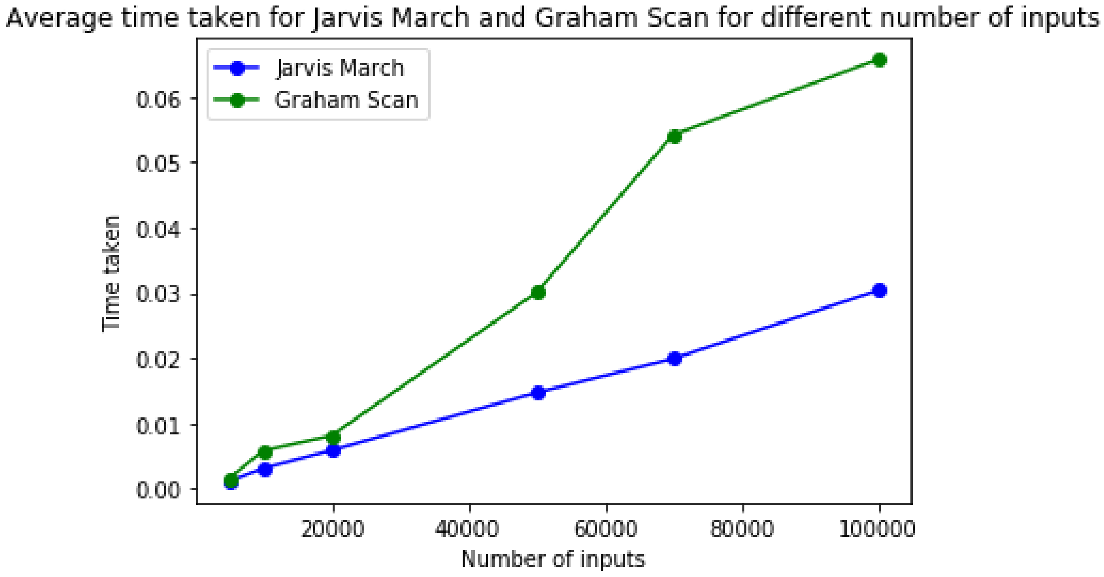
```
In [16]: plt.plot(n,jaravg,'-o',color = 'b')
         plt.title("Average time taken for Jarvis March for different number of inputs")
         plt.ylabel("Time taken")
         plt.xlabel("Number of inputs")
         plt.show()
```



```
In [17]: plt.plot(n,graavg,'-o',color = 'g')
         plt.title("Average time taken for Graham Scan for different number of inputs")
         plt.ylabel("Time taken")
         plt.xlabel("Number of inputs")
         plt.show()
```



```
In [18]: plt.plot(n,jaravg,'-o',color = 'b',label = 'Jarvis March')
         plt.plot(n,graavg,'-o',color = 'g',label = 'Graham Scan')
         plt.title("Average time taken for Jarvis March and Graham Scan for different number of
         plt.ylabel("Time taken")
         plt.xlabel("Number of inputs")
         plt.legend()
         plt.show()
```



0.2 Conclusion:-

0.2.1 The average time taken by Graham Scan is more than that taken by Jarvis March. This is going as expected because complexity of Graham Scan is O(nlogn) while complexity of Jarvis March is O(mn) where m is the number of edges in the convex hull. When, n is too large, m increases very slowly and is even less than log(n).