

```
!/usr/bin/env python3
(c) Dana Hughes 2021
```

python program to generate a Hilbert curve that fills in space using recursion methods

```
import the necessary modules
import string
import sys
import math
import matplotlib.pyplot as plot
import numpy as np
from itertools import chain
from matplotlib.backends.backend_pdf import PdfPages
```

```
track curve function
def hilbert(x, y, xi, xj, yi, yj, n):
    if n <= 0:
        return (x + (xi + yi)/2, y + (xj + yj)/2)
    else:
        a = [
            hilbert(x, y, yi/2, yj/2, xi/2, xj/2, n-1),
            hilbert(x + xi/2, y + xj/2, xi/2, xj/2, yi/2, yj/2, n-1),
            hilbert(x+xi/2+yi/2, y+xj/2+yj/2, xi/2, xj/2, yi/2, yj/2, n-1),
            hilbert(x+xi/2+yi, y+xj/2+yj, -yi/2, -yj/2, -xi/2, -xj/2, n-1),
        ]
        if n > 1:
            a = list(chain.from_iterable(a))
    return a
```

```
plotting function
__name__ == '__main__':
    n = input("Enter recursion: ")
    n = int(n)
    for i in range(n):
        if (n<1) or (n>8):
```

```
if (n == 0) or (n == 1):  
    False  
    sys.exit("error")  
# create the curve  
curve = hilbert(0.0, 0.0, 1.0, 0.0, 0.0, 1.0, n)  
x = [p[0] for p in curve]  
y = [p[1] for p in curve]  
plot.plot(x, y, 'r-', lw = 1)  
plot.savefig("hilbert#.pdf")
```

Enter recursion: 4



