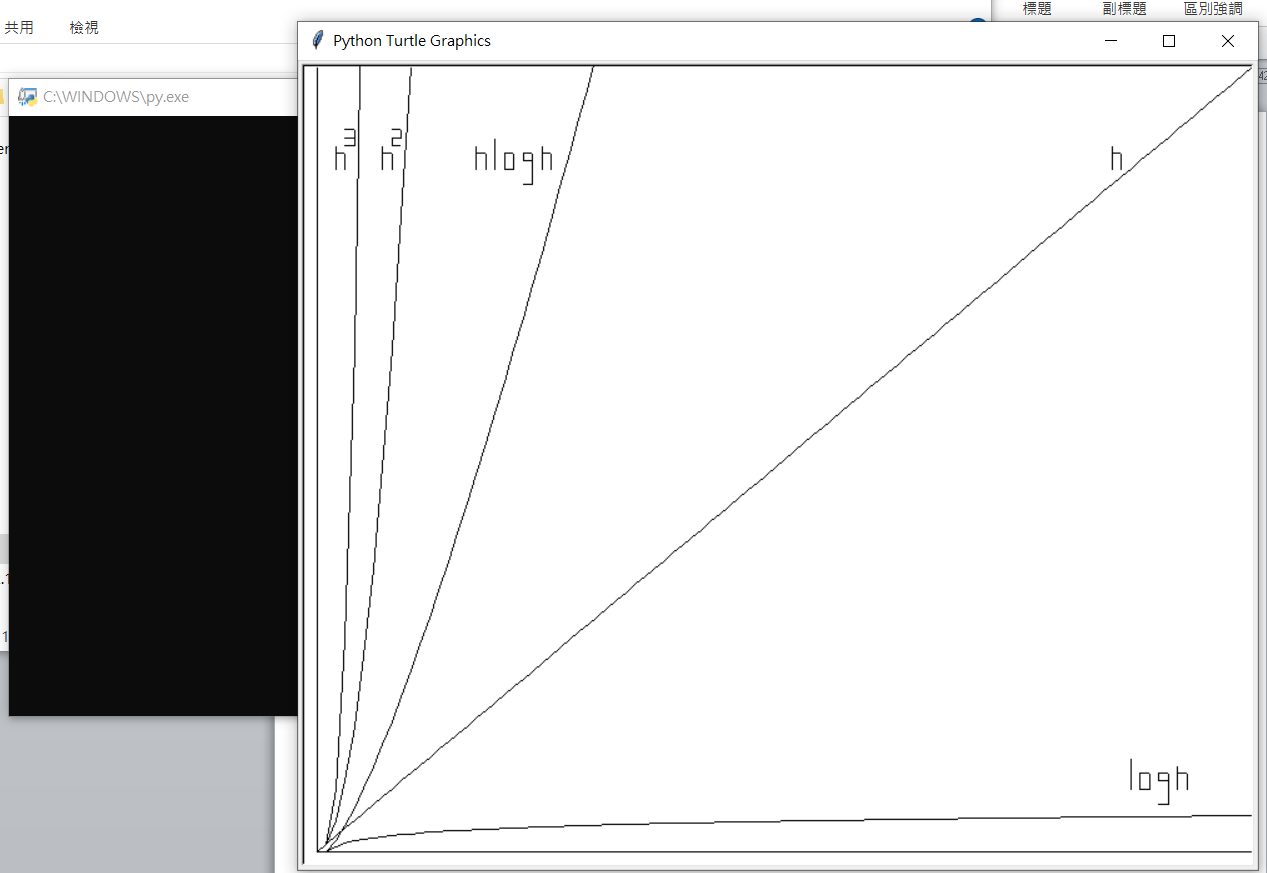
我的LineChart是用Turtle畫的

參考資料如下:

*http://yltang.net/tutorial/python/5/  
https://docs.python.org/zh-tw/3/library/turtle.html#turtle.pd*

按下line\_chart.py執行後會會出現下圖圖片，慢慢畫出圖片



Python程式碼如下:

利用各種函數，讓圖片畫出不同線條，次方函數在下面用for迴圈完成

*"""  
 參考網頁:  
 http://yltang.net/tutorial/python/5/  
 https://docs.python.org/zh-tw/3/library/turtle.html#turtle.pd  
"""*import math  
from turtle import Turtle, Screen  
  
screen = Screen()  
screen.setworldcoordinates(0, 0, 100, 100)  
  
emily = Turtle(visible=False)  
emily.speed(0)  
emily.forward(100)  
emily.forward(-100)  
emily.left(90)  
emily.forward(100)  
emily.penup()  
  
  
def straight\_line\_ud(x, num1, num2):  
 for y in range(num1, num2 + 1): # logn  
 emily.goto(x, y)  
 emily.pendown()  
 emily.penup()  
  
  
def straight\_line\_lr(y, num1, num2):  
 for x in range(num1, num2 + 1): # logn  
 emily.goto(x, y)  
 emily.pendown()  
 emily.penup()  
  
  
def char\_n(x\_min, y\_min):  
 straight\_line\_ud(x\_min, y\_min, y\_min + 3)  
 straight\_line\_lr(y\_min + 2, x\_min, x\_min + 1)  
 straight\_line\_ud(x\_min + 1, y\_min, y\_min + 2)  
  
  
def char\_2(x\_min, y\_min):  
 straight\_line\_lr(y\_min, x\_min, x\_min + 1)  
 straight\_line\_ud(x\_min + 1, y\_min, y\_min + 1)  
 straight\_line\_lr(y\_min + 1, x\_min, x\_min + 1)  
 straight\_line\_ud(x\_min + 1, y\_min + 1, y\_min + 2)  
 straight\_line\_lr(y\_min + 2, x\_min, x\_min + 1)  
  
  
def char\_3(x\_min, y\_min):  
 straight\_line\_lr(y\_min, x\_min, x\_min + 1)  
 straight\_line\_ud(x\_min, y\_min, y\_min + 1)  
 straight\_line\_lr(y\_min + 1, x\_min, x\_min + 1)  
 straight\_line\_ud(x\_min + 1, y\_min + 1, y\_min + 2)  
 straight\_line\_lr(y\_min + 2, x\_min, x\_min + 1)  
  
  
def char\_l(x\_min, y\_min):  
 straight\_line\_ud(x\_min, y\_min, y\_min + 4)  
  
  
def char\_o(x\_min, y\_min):  
 straight\_line\_ud(x\_min, y\_min, y\_min + 2)  
 straight\_line\_lr(y\_min, x\_min, x\_min + 1)  
 straight\_line\_ud(x\_min + 1, y\_min, y\_min + 2)  
 straight\_line\_lr(y\_min + 2, x\_min, x\_min + 1)  
  
  
def char\_g(x\_min, y\_min):  
 straight\_line\_ud(x\_min, y\_min + 3, y\_min + 4)  
 straight\_line\_lr(y\_min + 4, x\_min, x\_min + 1)  
 straight\_line\_ud(x\_min + 1, y\_min, y\_min + 4)  
 straight\_line\_lr(y\_min + 3, x\_min, x\_min + 1)  
 straight\_line\_lr(y\_min, x\_min, x\_min + 1)  
  
  
emily.penup()  
for x in range(1, 101): # logn  
 y = math.log(x)  
 emily.goto(x, y)  
 emily.pendown()  
  
emily.penup()  
  
# logn  
char\_l(87, 8)  
char\_o(88, 8)  
char\_g(90, 6)  
char\_n(92, 8)  
  
for x in range(0, 101): # n  
 y = x  
 emily.goto(x, y)  
 emily.pendown()  
  
emily.penup()  
  
# n  
char\_n(85, 87)  
  
for x in range(1, 101): # nlogn  
 y = x \* math.log(x)  
 emily.goto(x, y)  
 emily.pendown()  
 if y >= 100:  
 break  
  
emily.penup()  
  
# nlogn  
char\_n(17, 87)  
char\_l(19, 87)  
char\_o(20, 87)  
char\_g(22, 85)  
char\_n(24, 87)  
  
for x in range(0, 101): # n ^ 2  
 y = x \*\* 2  
 emily.goto(x, y)  
 emily.pendown()  
 if y >= 100:  
 break  
  
emily.penup()  
  
# n^2  
char\_n(7, 87)  
char\_3(8, 90)  
  
for x in range(0, 101): # n ^ 3  
 y = x \*\* 3  
 emily.goto(x, y)  
 emily.pendown()  
 if y >= 100:  
 break  
  
emily.penup()  
  
# n^3  
char\_n(2, 87)  
char\_2(3, 90)  
  
screen.exitonclick()