機器學習導論 HW1

1. 費氏數列（Fibonacci sequence）是一個由 0 和 1 開始，之後每個數值是前面兩個數之和的一個數列，下面我們列出費氏數列的前 6 個數

F_0	F_1	F_2	F_3	F_4	F_5
0	1	1 (= 0 + 1)	2 (= 1 + 1)	3 (= 1 + 2)	5 (= 2 + 3)

數學上，我們定義費氏數列如下：

$$F_i = \begin{cases} i & \text{if } i = 0,1 \\ F_{i-2} + F_{i-1} & \text{for } i \geq 2 \end{cases}$$

請寫一段程式，列出 $F_{23}, F_{24}, \cdots, F_{32}$。

```
def Fibonacci(i):
    if i == 0 or i == 1:
        answer = i
    elif i >= 2:
        answer = Fibonacci(i-2) + Fibonacci(i-1)
    return answer

for i in range(23, 33):
    print('F%d: %d'%(i, Fibonacci(i)))
```

```
F23: 28657
F24: 46368
F25: 75025
F26: 121393
F27: 196418
F28: 317811
F29: 514229
F30: 832040
F31: 1346269
F32: 2178309
```

---

2. 請先建立一個串列(list) a，包含下列元素

4	−2	7	10	2	−9	100	3	15

再用**串列生成式(list comprehension)** 語法，產生一個包含串列 a 中奇數元素平方的串列。

```
a = [4, -2, 7, 10, 2, -9, 100, 3, 15]
answer = [i**2 for i in a if i%2!=0]
answer
```

```
[49, 81, 9, 225]
```

3. 草莓站是個只有一個月台的火車站，為了提升服務品質，減少等待時間，必須妥善安排列車進站的順序。假設有下表兩輛列車，都在時間點 1 到達草莓站外，如果安排列車 A 先進站，則列車 B 必須等 A 停留 5 個時間點讓乘客上下車結束並離站後，才能進站，因此總等待時間是 5；而如果安排 B 先進站，則總等待時間為 8。

列車編號	到達站外時間	需要停留時間	等待時間	
			A 先進站	B 先進站
A	1	5	0	8
B	1	8	5	0
總等待時間			**5**	**8**

(1) 假設有 5 輛列車在同一個時間點到達，請安排進站順序，讓總等待時間最短，並算出總等待時間。

列車編號	到達站外時間	需要停留時間
MA	10	8
MB	10	2
MC	10	4
MD	10	7
ME	10	5

```
train = ['MA', 'MB', 'MC', 'MD', 'ME']
time = [8, 2, 4, 7, 5]
wait_time = 0

# wait_time = 4*time[0]+3*time[1]+2*time[2]+1*time[3]
sort = sorted([i for i in zip(train, time)], key=lambda k: k[1]) #用第二個元素作排序
for i, s in enumerate(sort):
    wait_time += (4-i)*s[1]
    print(f'第{i+1}順位進站：{s[0]}')
print(f'\n總共等待{wait_time}分鐘')
```

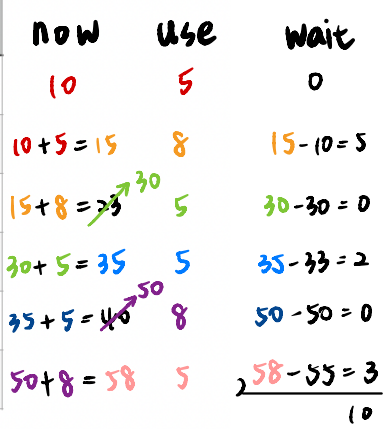
('MA', 8)
('MB', 2)
⋮

第1順位進站：MB
第2順位進站：MC
第3順位進站：ME
第4順位進站：MD
第5順位進站：MA

總共等待37分鐘

(2) 假設會停靠草莓站的列車有兩種 MM 和 MX，分別需要停留 5 和 8 個時間點，假設當日的時間表如下，請安排進站順序，讓總等待時間最短，並算出總等待時間。

列車編號	到達站外時間	需要停留時間
MM101	10	5
MX101	10	8
MM102	30	5
MM103	33	5
MX102	50	8
MM104	55	5




```
train = ['MM101', 'MX101', 'MM102', 'MM103', 'MX102', 'MM104']
arrive = [10, 10, 30, 33, 50, 55]
time = [5, 8, 5, 5, 8, 5]
now_time = 0
use_time = 0
wait_time = 0

sort = sorted([(train[i], arrive[i], time[i]) for i in range(len(train))], key=lambda k: (k[1], k[2]))
for i, s in enumerate(sort):
    print(f'第{i+1}順位進站：{s[0]}')

for i in sort:
    if now_time == 0:
        now_time += i[1]
        use_time = i[2]
    else:
        now_time = max(now_time+use_time, i[1])
        wait_time += now_time-i[1]
        use_time = i[2]
print(f'\n總共等待{wait_time}分鐘')
```

第1順位進站：MM101
第2順位進站：MX101
第3順位進站：MM102
第4順位進站：MM103
第5順位進站：MX102
第6順位進站：MM104

總共等待10分鐘