

# 用GA來解 最短路徑問題

# 問題

- 假如在一張地圖上有 $N$ 個點，我可能在地圖上的任意一個點上，我該如何走才能走出最短的路徑，並且經過每一個城市？

# DNA編碼

- 用走過的城市順序
- 問題：交配/突變時重複已走過的城市

# Solution

```
def crossover(self, parent, pop):  
    if np.random.rand() < self.cross_rate:  
        # select another individual from pop  
        i_ = np.random.randint(0, self.pop_size, size=1)  
        cross_points = np.random.randint(0, 2, self.DNA_size).astype(  
            np.bool) # 產出一個與DNA大小相同的True False 陣列  
        keep_city = parent[~cross_points]  
        # # 陣列中是 False 的 Index 會被加到新的DNA內  
        swap_city = pop[i_, np.isin(  
            pop[i_].ravel(), keep_city, invert=True)]  
        # 在pop內把不在 keep_city 內的元素放到 swap_city  
        parent[:] = np.concatenate((keep_city, swap_city))  
        # 連結 產生DNA  
    return parent
```

# Fitness

- 把全部的點連起來算距離，距離越短fitness越高
- 雨天
- 優先權

# 雨天

- 雨天隨機產生，若遇到雨天 總距離+0.5%
- 雨天DNA fitness +6.6%

# 優先權

- 點點產生後產生一組跟點點數量一樣的陣列，裡面放0~3的數字（表示優先權）
- 用一個積分來計算優先權的分數（積分 += 優先權）
- 每做一輪動作全部的優先權-1，減到0就不算積分了

# Solution

```
def priorityCalc(self, inputArray, priority):  
    priority_copy = priority.copy() # copy 一份優先權清單  
    score = 0 # 初始化積分  
    for i in inputArray:  
        if priority_copy[i] > 0: # 假如積分不小於零就加  
            score = score + priority_copy[i]  
        priority_copy[:] = priority_copy[:] - 1 # 跑過一輪後全部的優先權 -1  
    return score
```



結果 = DEMO

# 遇到的問題

- 起點永遠都被選在邊邊（中間的點可能不太會被選為起點）
- 無法自訂起點

感謝大家