

資工碩一

108062566

陳法佑

我設計了兩個 mapper、reducer，第一個 mapper 如附圖。假設是 M、N 都是 $(500 * 500)$ 的矩陣、輸出為 C 矩陣，那麼當 M 矩陣 $(2, 0)$ 的元素經過這個 mapper，它的 key 會存成 $(2, 0, 0) \sim (2, 499, 0)$ ，如果是 N 矩陣 $(2, 0)$ 的元素，則會存成 $(0, 0, 2) \sim (499, 0, 2)$ 。Key 的前兩維表示目前這個 value 貢獻在 C 矩陣的位置，第三維表示目前 value 應該要對應到另一個矩陣的哪個位置，因為會有 1000 個 value（M、N 矩陣各 500 個）貢獻到 C 矩陣的相同位置，所以才需要加上這個維度。

```
def mapper1(line):
    matrix_name, row, col, num = line.split(",")
    row, col, num = int(row), int(col), int(num)
    mapList = []
    for idx in range(MATRIX_SIZE):
        if matrix_name == 'M':
            key = (row, idx, col)
            value = num
        else:
            key = (idx, col, row)
            value = num
        mapList.append([key, value])
    return mapList
```

接著 reducer1 很簡單，就是把同個 key 的 value 相乘，結果會使每個 C 矩陣位置會存有 500 個值，原本 1000 個值變為 500 是因為 reducer1 將兩個值相乘的緣故。

```
def reducer1(x, y):
    return x * y
```

接著就是要把對應到 C 矩陣同個位置的 500 個值加起來，所以先透過 mapper2 把 reducer1 後每個 key（包含 row, col, idx）的 row, col 抓出來，並存入新的 key，value 則不變。

```
def mapper2(x): # x => ((0, 0, 0), 10)
    row, col, idx = x[0]
    key = (row, col)
    value = x[1]
    return [(key, value)]
```

最後再把同個 key 的值加起來就是 C 矩陣各個位置矩陣相乘的結果。

```
def reducer2(x, y):
    return x + y
```

結果（另外包含 500output.txt）

```
0,0,1224217
0,1,1209205
0,2,1172528
0,3,1150746
0,4,1152989
0,5,1143435
0,6,1192906
0,7,1163613
0,8,1146928
0,9,1270345
0,10,1164367
0,11,1197043
0,12,1197989
0,13,1132939
0,14,1169192
0,15,1196336
0,16,1170027
0,17,1226865
0,18,1246305
0,19,1233307
0,20,1193495
0,21,1144991
0,22,1188082
0,23,1136498
0,24,1171027
0,25,1257640
0,26,1172161
0,27,1171267
0,28,1158241
0,29,1161176
0,30,1207314
0,31,1191679
0,32,1212365
0,33,1151200
0,34,1260592
0,35,1166960
0,36,1177580
0,37,1228129
0,38,1128280
0,39,1197121
0,40,1174699
0,41,1183543
0,42,1208107
0,43,1153855
0,44,1152943
0,45,1188817
0,46,1154121
0,47,1159208
0,48,1231126
0,49,1153542
0,50,1158402
0,51,1218339
```