Data Structure - Homework 2: Transpose

A1115513 劉沛辰

三種轉置方式和運行時間:

```
5 6
0 0 1
0 2 1
1 1 2
2 0 4
2 2 3
^Z
Traditional 2-dimensional array:
1 0 4 0 0 0
0 2 0 0 0 0
1 0 3 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0

1 0 1 0 0
0 2 0 0 0
4 0 3 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
traditional 2-dimensional array execution time: 10 milliseconds

Fast Transpose:
0 0 1
0 2 4
1 1 2
2 0 1
2 2 3
Transpose execution time: 5 milliseconds

Fast Transpose:
0 0 1
0 2 4
1 1 2
2 0 1
2 2 3
Fast Transpose execution time: 0 milliseconds
```

各種測資的時間(助教給的.in):

7_9:

```
traditional 2-dimensional array execution time: 12 milliseconds

Transpose execution time: 5 milliseconds

Fast Transpose execution time: 0 milliseconds
```

15_12:

```
traditional 2-dimensional array execution time: 19 milliseconds

Transpose execution time: 12 milliseconds

Fast Transpose execution time: 1 milliseconds
```

60_74:

```
traditional 2-dimensional array execution time: 115 milliseconds

Transpose execution time: 610 milliseconds

Fast Transpose execution time: 36 milliseconds
```

100_100:

```
traditional 2-dimensional array execution time: 398 milliseconds

Transpose execution time: 2990 milliseconds

Fast Transpose execution time: 104 milliseconds
```

256_512:

```
traditional 2-dimensional array execution time: 2706 milliseconds

Transpose execution time: 85494 milliseconds

Fast Transpose execution time: 1091 milliseconds
```

720_850:

```
traditional 2-dimensional array execution time: 12343 milliseconds

Transpose execution time: 853105 milliseconds

Fast Transpose execution time: 7716 milliseconds
```

第一種演算法的時間複雜度為:O(col*row)

第二種演算法的時間複雜度為:O(col*輸入行數)

第三種演算法的時間複雜度為:O(輸入行數)

經過上面 testcase 的統計，雖然在陣列大小很小時的時間差距不大，但是到了 720_850 時，第一種演算法比第二種快了 71 倍，第三種比第二種快了 1218 倍，可見 O(n^2)和 O(n)的速度差距。