Algorithm Task

10 - Diagonal Difference

Team Number: 165

	ID	Full Name
1	20210159	إسلام مجدي محمد الصفي علي الجزار
2	20210052	أحمد رفعت كامل أحمد عفيفي
3	20211104	يوسف محمد نادي حسن مرسي
4	20210081	أحمد علي إبراهيم سليمان حرب
5	20210057	أحمد سعدالدين محمد علي المراكبي
6	20210050	أحمد رضا شعبان عبدالوهاب

1- Non-recursive:

1.1- Pseudocode:

```
ALGORITHM DiagonalDifference(n, arr) {
        primaryDiagonal <- 0
        secondaryDiagonal <- 0
        for i <- 0 to n – 1 do
                primaryDiagonal <- primaryDiagonal + arr[i][i]
                secondaryDiagonal <- secondaryDiagonal + arr[i][n - i - 1]
        return abs(primaryDiagonal - secondaryDiagonal)
}
```

1.2- Analysis:

$$\sum_{i=0}^{n-1} 1 = n - 1 - 0 + 1 = n$$

So, Time Complexity is Θ(n)

```
3
11 2 4
4 5 6
10 8 -12
15
Process returned 0 (0x0)   execution time : 19.548 s
Press any key to continue.
```

2- Another Non-recursive

2.1- Pseudocode:

```
ALGORITHM DiagonalDifference(n, arr) {
        primaryDiagonal <- 0
        secondaryDiagonal <- 0
        for i <- 0 to n – 1 do
                for j <- 0 to n – 1 do
                        if i = j
                                primaryDiagonal <- primaryDiagonal + arr[i][j]
                        if i = n - j - 1
                                secondaryDiagonal <- secondaryDiagonal + arr[i][j]
        return abs(primaryDiagonal - secondaryDiagonal)
}
```

2.2- Analysis:

$$\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} 1 = \sum_{i=0}^{n-1} n - 1 - 0 + 1 = \sum_{i=0}^{n-1} n = n^2$$

So, Time Complexity is $\Theta(n^2)$

```
3
11 2 4
4 5 6
10 8 -12
15
Process returned 0 (0x0)   execution time : 6.401 s
Press any key to continue.
```

3- Recursive:

3.1- Pseudocode:

```
ALGORITHM DiagonalDifference(n, arr, i) {
        If (i = n)
                return 0
        return arr[i][i] – arr[n – i – 1] + DiagonalDifference(n, arr, i + 1)
}
```

3.2- Analysis:

$$
\begin{aligned}
T(n) &= T(n-1) + 1 \\
T(n-1) &= T(n-2) + 1 \\
T(n) &= (T(n-2) + 1) + 1 \\
&= T(n-2) + 2 \\
T(n-2) &= T(n-3) + 1 \\
T(n) &= (T(n-3) + 1) + 2 \\
&= T(n-3) + 3 \\
T(n) &= T(n-k) + k \\
n - k = 1 &\implies k = n - 1 \\
T(n) &= T(1) + n - 1 \\
&= 1 + n - 1 \\
&= n
\end{aligned}
$$

So, Time Complexity is Θ(n)

```
3
11 2 4
4 5 6
10 8 -12
15
Process returned 0 (0x0)   execution time : 18.108 s
Press any key to continue.
```

3- Comparison

ALGORITHM	Time Complexity		
	Best Case	Average Case	Worst Case
Non-recursive	$\Omega(n)$	$\Theta(n)$	$O(n)$
Another Non-Recursive	$\Omega(n^2)$	$\Theta(n^2)$	$O(n^2)$
Recursive	$\Omega(n)$	$\Theta(n)$	$O(n)$

First Non-recursive Algorithm is better than another one, because it uses just 1 for loop, So Its Time Complexity is lower

And both of the first Non-recursive, and Recursive Algorithm have the same Time Complexity, but as Performance Non-recursive is better