Diagonal Difference

1- Non-recursive:

1.1- Implementation:

```c
#include <stdio.h>
#include <math.h>
int diagonalDifference(int n, int arr[][n]) {
    int primaryDiagonal = 0, secondaryDiagonal = 0;
    for (int i = 0; i ⩽ n - 1; i++) {
        primaryDiagonal = primaryDiagonal + arr[i][i];
        secondaryDiagonal = secondaryDiagonal + arr[i][n - i - 1];
    }
    return abs(primaryDiagonal - secondaryDiagonal);
}
int main() {
    int n;
    scanf("%d", &n);
    int arr[n][n];
    for (int i = 0; i ⩽ n - 1; i++) {
        for (int j = 0; j ⩽ n - 1; j++) {
            scanf("%d", &arr[i][j]);
        }
    }
    printf("%d", diagonalDifference(n, arr));
}
```

1.2- Documentation:

```
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)
}
```

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

So, Time Complexity is Θ(n)

2- Recursive:

2.1- Implementation:

```c
#include <stdio.h>
#include <math.h>
int diagonalDifference(int n, int arr[][n], int i) {
    if (i == n) {
        return 0;
    }
    return arr[i][i] - arr[i][n - i - 1] + diagonalDifference(n, arr, i + 1);
}
int main() {
    int n;
    scanf("%d", &n);
    int arr[n][n];
    for (int i = 0; i <= n - 1; i++) {
        for (int j = 0; j <= n - 1; j++) {
            scanf("%d", &arr[i][j]);
        }
    }
    printf("%d", abs(diagonalDifference(n, arr, 0)));
}
```

2.2- Documentation:

ALGORITHM DiagonalDifference(n, arr, i) {

If (i = n)

return 0

return arr[i][i] – arr[n – i – 1] + DiagonalDifference(n, arr, i + 1)

}

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

So, Time Complexity is Θ(n)