**Diagonal sums difference**

|  |  |
| --- | --- |
| Team member ID | Team member name |
| 20220422 | **محمد عماد الدين محمود حامد** |
| 20220403 | **محمد سيف النصر عارف رمضان** |
| 20220546 | **وجيه مصطفي وجيه اسعد يوسف** |
| 20220221 | **سيف شريف خضر صالح** |
| 20220334 | **فرج سلامة فرج حسن** |
| 20220384 | **محمد ايمن محمد حسني** |
| 20220579 | **يوسف وائل عطالله موسي محمود** |

**Non-recursive algorithm :**

**Pseudocode :**

**A screenshot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generatedImplementation :**

**Analysis & complexity**

A screenshot of a computer program

Description automatically generated

**Detailed T(n):   
T(n) = (n+1)2+n2+2n+8**

**T(n) = 2n2+4n+9**

**T(n) = O(max(2n2 , 4n , 9)**

**T(n) = O(n2)**

**Shorten T(n):**

**T(n) = (n+1)2+ O(1)**

**T(n) = (n2 + 2n + 2) + O(1)**

**T(n) = O(max(n2 , n , 1)) = O(n2)**

**Best case of T(n) = O(n2)**

**worst case of T(n) = O(n2)**

**Recursive algorithm :**

**A screenshot of a computer program

Description automatically generatedPseudo code :**

**A screen shot of a computer program

Description automatically generatedImplementation :**

**A screen shot of a computer program

Description automatically generatedAnalysis & complexity :**

**c = O(1)**

**Recursion Analysis**

**T(n) = T(n-1) + c and T(1) = O(1)**

**T(n) = T(n-1) + O(1)**

**where :T(n-1) = T(n-2) + O(1)**

**T(n) = T(n-2) + O(1) + O(1)**

**Where T(n-2) = T(n-3) + O(1)**

**T(n) = T(n-3) + O(1) + O(1) + O(1)**

**T(n) = T(n-3) + 3 O(1)**

**T(n) = T(n-k) + k O(1)**

**T(n-k) = T(1) = O(1), when k = n - 1**

**T(n) = T(n-n-1) + (n-1) O(1)**

**T(n) = T(1) + n**

**T(n) is O(n)**

**Analysis of the whole code :**

**T(n) = O(n) + (N + 1) \* (N+1)**

**T(n) = O(n) + N2 + 2n + 2**

**T(n) is max(N2 , n , 2)**

**T(n) is O(N2)**

**Best case is O(n2)**

**Worst case is O(n2)**