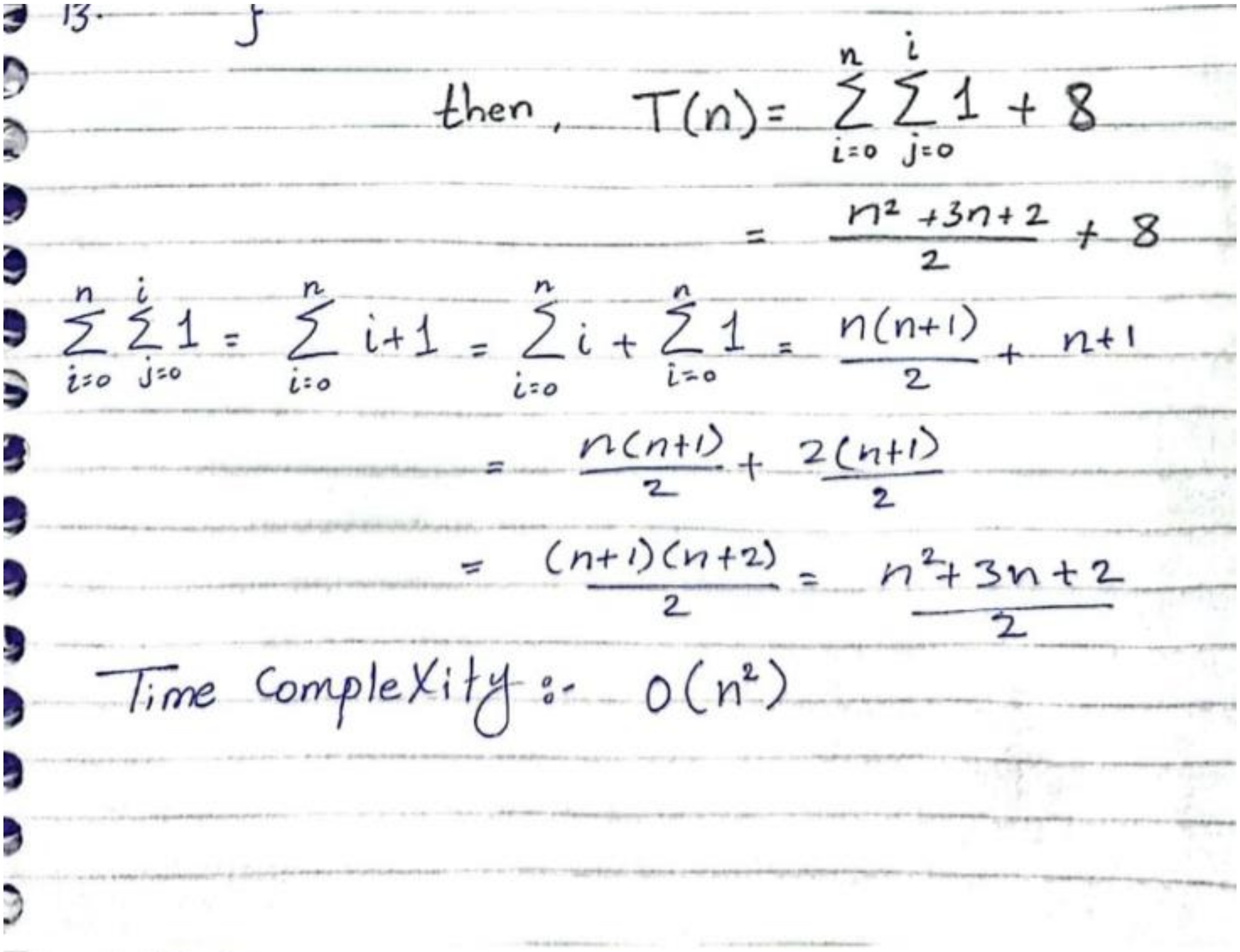
```
Algorithm countdistinct(A,n)
{
  distinctcount:=0 ---->1
  for i:=0 to n do ---->sum i=0 to n
         Isdistinct:=true ---->1
         for j:=0 to I do ---->sum j=0 to i
              If A[i] == A[j] then ---->1
                 Isdistinct:=false ---->1
                 Break ---->1
        If isdistinct==true then ---->1
            distinctcount=distinctcount+1 ---->1
   return distinctcount ---->1
}
```

==

ANALYSIS



13. }

then, $T(n) = \sum_{i=0}^{n} \sum_{j=0}^{i} 1 + 8$

$= \frac{n^2+3n+2}{2} + 8$

$\sum_{i=0}^{n} \sum_{j=0}^{i} 1 = \sum_{i=0}^{n} i+1 = \sum_{i=0}^{n} i + \sum_{i=0}^{n} 1 = \frac{n(n+1)}{2} + n+1$

$= \frac{n(n+1)}{2} + \frac{2(n+1)}{2}$

$= \frac{(n+1)(n+2)}{2} = \frac{n^2+3n+2}{2}$

Time CompleXity :- $O(n^2)$

===

TIME COMPLEXITY

Best-case time complexity is O(n)

20.Worst-case time complexity is O(n^2)

21.Average-case time complexity is O(n^2).