1. WORST CASE TIME COMPLEXITY for ElectionCount class using Direct Addressing through Arrays:
   1. ADD – O(1) [as index is known in advance by using voter\_id]
   2. FIND – O(1) [as index is known in advance by using voter\_id]
   3. COUNT – O(1) [as index is known in advance by using candidate\_id]
2. WORST CASE TIME COMPLEXITY for ElectionCount class using Direct Addressing through HashTable (mapped using voter\_id and candidate\_id as key):
   1. ADD – O(1) [index is known in advance using voter\_id as key]
   2. FIND – O(1) [index is known in advance using voter\_id as key]
   3. COUNT – O(1) [index is known in advance using candidate\_id as key]

Example: for following list:

151020 130

151021 135

151022 132

151023 135

151024 130

151025 135

151026 130

151027 135

151028 132

151029 130

151030 135

151031 135

100000 100

999999 999

1. In Array

ADD(151020,130) for Array will add at index 251020 directly

FIND(151020) for Array will directly search at index 251020

COUNT(135) will search directly at index 35 and retrieve the count as 6.

1. In HashTable

ADD(151020,130) will add directly using keys 151020 and 130

FIND(151020) will directly retrieve candidateId using key 151020

COUNT(135) will directly retrieve size of Vote ArrayList against key 135