1. Algorithm1 executes in O(n^2).

It means that the algorithm takes time which is quadratic of its input array. It starts to visit each element present in the given array. It does so with a for loop having an index variable i. The complexity here will be O(n).For each element, it again visits all other elements following it. This block also has complexity O(n), as it uses a nested for loop having an index variable j.Inside the nested for loop, algorithm checks whether sum of current (arr[i]) and next element (arr[j]) is 0. If it is true - then algorithm prints messages into a console. This step has constant complexity O(1).As soon as an algorithm performs two nested for loops - its whole runtime complexity will be considered as O(n^2).

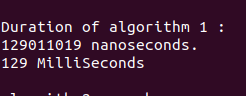
Algorithm2 executes in O(n).

It means that the algorithm takes runtime which is linearly of its input array. The algorithm creates a hashset to store all those visited elements. This step takes a constant amount of time which is O(1). To initialise the index i and j also takes constant amout of time which is O(1). The algorithm terminates when the two indexes become equal. While looping the algorithm does the following, it calculates the sum of the current elements at i and j and this process takes constant amount of time. It has the condition statements to check if the sum of the current elements are equal to zero and if the elements have been visited or not. If the elements have a sum equal to zero then they are added to the set. Therefore this step takes O(1). Hence the algorithm stops when the two indexes i and j become equal, it will iterate n times which produces a runtime time complexity of O(n).

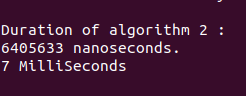
2. Algorithm2 is more efficient than one. Since algorithm2 executes in O(n) means that the time is proportional to the input size.

3. The actual time for the algorithms supports that Algorithm2 runs faster than Algorithm1, please see the screenshots below.

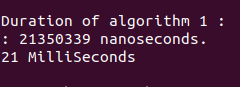
**Screenshot of algorithm 1 running the Input1.txt dataset.**



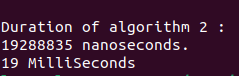
**Screenshot of algorithm 2 running the Input1.txt dataset.**

****

**Screenshot of Algorithm 1 running the Input2.txt dataset.**

****

**Screenshot of Algorithm 2 running the Input2.txt dataset**

****