1)

Answer: The algorithm in the original program,

1. Reads the file line by line and then saves it in a list.
2. Each i element in the list is compared with remaining n-1 elements.
3. The outer for loop runs for n number of times and the inner for loop runs for n number of time.
4. Inside the nested for loop the two values are read form the lists. The value corresponds to the ith index of the first for loop and is compared with the 0-n indexed values of the list.

Reason for reporting more than once is,

Both the for loops are running from 0 to n. So each duplicate found is compared for its duplicate. If list[y] is the duplicate of list[x], then list[y] is reported as duplicate of list[x] and list[x] is reported as duplicate of list[y]. But the correct result is to report only one duplicate that is list[y].

This is reason why the duplicates are reported more than once.

2)

Answer: The Big-O running time of the original program is O(n^3)

Calculating Big-O for the original program:

1. The outer for loop runs for n times.
2. The inner for loop runs for n times.
3. The get method of the linked list takes n times.
4. There are two get methods but the Big O for them gets added and time will be counted as 2n.
5. The Big-O for the algorithm is n\*n\*2n = 2n^3
6. Which is equal to O(n^3)

Answer: The algorithm in the new program is,

1. Reads the file line by line.
2. Checks for duplicate in the hashTable. If the duplicate is present then the value is printed as duplicate.
3. To check for the duplicate it doesn’t need to traverse the whole hashTable. The hashTable saves the values based on the key values and key values are calculated based on the data value you are saving.
4. The key values will then serve as the index of the Array list(which makes up the hashTable).
5. To find any element in the hashTable we will calculate its key value and directly access the value with constant time. As the get element from the arrayList will give us O(1).
6. Adds the value it to hashTable.

Answer: The Big-O running time for the new program is O(n)

Calculating Big-O for new algorithm,

1. The while loop runs for n times.
2. Check duplicate takes constant running time. As it calculates the key value and get the values present in the arrayList at that index. If the duplicate is found it is printed in the console else it will not.
3. Add the value to the hashTable takes constant running time. Again to add the element it calculates the hash Key value and makes it as the index of arrayList at which the value is added.
4. As the hashTable works on the hash key concept the value are not pushed upwards or downwards.