Given an unsorted array of n items (called A), it is possible to implement an algorithm that returns the first duplicate item (by index) contained within the array (or null if there are no duplicates). One possible implementation is given below.
Algorithm:

1. Initialize a new empty container (called container B)
2. Loop through array A beginning at index 0
3. Check if container B contains the current element of A

-If B contains the element then we have found the first duplicate and return it
-If B does not contain the element then we add the element to container B

4. If the loop terminates, there were no duplicates in array A so we return null

Your task is to determine the effects of utilizing different data structures for container B in the above algorithm in regards to worst case runtime. Find the worst case runtime for the algorithm if container B is:
(i) An unsorted array
(ii) A sorted linked list
(iii) A hashmap
Which data structure is the optimal one to use for this implementation in terms of runtime?
Write the pseudocode for the algorithm using the steps laid out above, the data structure with the optimal worst case runtime, and any primitives of that data type (e.g. push, pop for stack etc.).