9)

a) La complejidad del algoritmo es O(n)

b) Lo que hace este algoritmo es implementar un set a partir de una lista. Un set es una colección de tipos de datos que puede ser mutable, iterable y que además no tiene ningún elemento repetido.

c)

Class set: Esta clase es la que se va a encargar del Set, allí se van a contener todos los métodos relacionados a este propósito, como lo es el método “add” o “remove”.

Método \_\_init\_\_: Este es el constructor y lo que va a hacer es crear un atributo \_theElements\_“que va a ser una lista

método \_\_len\_\_: Este se va a encargar de devolver el numero de elementos de la lista (“\_theElements\_”).

Método \_\_contains\_\_ : Este método va a recibir como parámetro un elemento y lo que va a hacer es determinar si el elemento se encuentra en el set. Esto lo hace usando otro método que encuentra el índice del elemento a buscar y luego en el caso de que este índice sea menor a la longitud del set y que el elemento en ese índice sea igual al buscado entonces devolverá true y en caso contrario devolverá un false.

Método add: Este método se encarga de añadir un elemento al set que se ingresa como parámetro. Primero verifica que este elemento no se encuentre ya en el set y en caso de que no se encuentre lo añadirá, y en caso de que ya halla un elemento igual no lo añadirá.

Método remove: Este método primero revisa que el elemento ingresado como parámetro este en el set porque si no arrojara error, y en caso de que el elemento se encuentre, lo sacara del set.

Método isSubsetOf: Este método se encarga de verificar si un set ingresado por parámetro es subconjunto de otro. Esto lo hace comparando que cada elemento se encuentre en el otro set.

Método \_\_iter\_\_: Este método se encarga de devolver un iterador para poder recorrer los ítems del set.

Método \_findPosition: Este método recibe un elemento por parámetro y lo que hace es que usando una búsqueda binaria, retorna la posición en la que se encuentra el elemento en el set.