

RESULTADO DE LA CORRECCIÓN: **APROBADO-**

OBSERVACIONES GENERALES

No permite trabajar con conjuntos vacíos. Errores en la determinación de la complejidad. Da error la función que usan para borrar la pantalla.

```

+-----+
| 1. Probar funciones del punto 2
| 2. Probar funciones del punto 3
| 3. Probar funciones del punto 4 ( Transitividad )
| 4. Probar funciones del punto 5 ( Diferencia simetrica )
| 5. Probar funciones del punto 6 ( Determinar subconjunto )
| 6. Probar funciones del punto 7 ( Determinar subconjunto total - parcial )
| 7. Probar funciones del punto 8 ( Determinar si son iguales )
| 8. Ver complejidades
| 9. Salir
+-----+
Ingrese una opcion:
6
"clear" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
"clear" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
Ingrese cuantos elementos queres agregar al conjunto: 0
Ingrese un numero valido:

```

```

Complejidad de los ejercicios dados
-----
Complejidad ejercicio 6 (Ver si un conjunto es subconjunto del otro):
Implementado con listas y utilizando lista con arreglos la complejidad es O(n)
Implementado con listas y utilizando lista con punteros o cursores la complejidad es O(n^2)
Implementado con AVL y la complejidad es O(n^2) si se utiliza listas con punteros o cursores ya que la funci|n recupera
del TAD, utiliza l_agregar
Implementado con AVL y la complejidad es O(n) si se utiliza listas con arreglos ya que la funci|n recupera del TAD, uti
liza l_agregar
-----
Complejidad ejercicio 7 (Ver si un conjunto es subconjunto total o parcial):
Implementado con listas y utilizando lista con arreglos la complejidad es O(n)
Implementado con listas y utilizando lista con punteros o cursores la complejidad es O(n^2)
Implementado con AVL y la complejidad es O(n^2) si se utiliza listas con punteros o cursores ya que la funci|n recupera
del TAD, utiliza l_agregar
Implementado con AVL y la complejidad es O(n) si se utiliza listas con arreglos ya que la funci|n recupera del TAD, uti
liza l_agregar
-----
Complejidad ejercicio 8 (Ver si un conjunto es igual al otro sin importar la posicion):
Implementado con listas y utilizando lista con arreglos la complejidad es O(n)
Implementado con listas y utilizando lista con punteros o cursores la complejidad es O(n^2)
Implementado con AVL y la complejidad es O(n^2) si se utiliza listas con punteros o cursores ya que la funci|n recupera
del TAD, utiliza l_agregar
Implementado con AVL y la complejidad es O(n) si se utiliza listas con arreglos ya que la funci|n recupera del TAD, uti
liza l_agregar
-----

```

En el ejercicio 6 para verificar que los conjuntos sean distintos, una vez determinada la inclusión de A en B, usan diferencia simétrica pudiendo usar la cantidad de elementos de B. En el ejercicio 7 si ningún conjunto es subconjunto parcial de otro no muestra ningún mensaje.

GRUPO 1 – Correcciones Trabajo Práctico: CONJUNTOS

```
El elemento se agrego correctamente
Ingrese la clave: 2
El elemento se agrego correctamente
Ingrese la clave: 3
El elemento se agrego correctamente
"clear" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
Ingrese cuantos elementos quieres agregar al conjunto: 3
Ingrese la clave: 4
El elemento se agrego correctamente
Ingrese la clave: 5
El elemento se agrego correctamente
Ingrese la clave: 6
El elemento se agrego correctamente
"clear" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
Ingrese cuantos elementos quieres agregar al conjunto: 3
Ingrese la clave: 1
El elemento se agrego correctamente
Ingrese la clave: 4
El elemento se agrego correctamente
Ingrese la clave: 7
El elemento se agrego correctamente

Presione Enter para continuar..._
```