

Algoritmos y Estructuras de Datos
Lic. en Agroinformática - Lic. en Bioinformática
2024

- 1) ¿Qué se obtiene al ejecutar el siguiente código? Justifique

```
l1 = ["a", "b", "c", "d"]  
l2 = l1  
l1[2] = True  
print(l2)
```

- 2) La función "id" devuelve la posición de memoria donde está alojada una variable. Compruebe la respuesta del punto anterior utilizándola.
- 3) Implemente las clases Nodo y ListaSimplementeEnlazada (o LSE). La clase LSE debe contener los siguientes atributos y métodos:
- a) Una variable con el tamaño de la lista y un puntero al comienzo de la lista.
 - b) Un método para obtener el tamaño
 - c) Un método para saber si la lista está vacía.
 - d) Un método para imprimir el contenido de la lista.
 - e) Un método para agregar al comienzo y otro para agregar al final.
 - f) Un método para quitar del comienzo.
- 4) ¿Cómo podría obtener el anteúltimo elemento de una LSE recorriendo la lista?
- 5) Implemente un método para eliminar el último elemento y otro para eliminar todas las instancias de un valor.
- 6) Implemente la función `__getitem__` que permita acceder a un elemento de la lista utilizando corchetes (por ejemplo, `lse[2]` para acceder al tercer elemento).
- 7) Implemente una función para agregar y para eliminar elementos en posiciones determinadas.
- 8) Implemente la clase ListaDoblementeEnlazada (LDE). Qué funcionalidades deberían ser agregadas y cuales modificadas con respecto a la clase LSE?
- 9) La función `now` del módulo `datetime` permite obtener la hora actual. Calculando el tiempo antes y después de correr un bloque de código es posible determinar el tiempo que demandó el código.

```
from datetime import datetime  
antes = datetime.now()
```

```
# lista de sentencias
despues = datetime.now()
print("tiempo_transcurrido: ", despues - antes)
```

Utilice esta estrategia para determinar cuánto tiempo demanda agregar 10 elementos en el medio de un *array* y una lista enlazada de 1000, 10000 y 100000 elementos.

10) A partir de la siguiente secuencia de operaciones, que resultados se obtendrían a partir de una pila vacía y de una cola vacía?

- | | | | |
|----|-------------|----|-------------|
| a) | agregar "5" | i) | eliminar |
| b) | agregar "3" | j) | agregar "7" |
| c) | eliminar | k) | agregar "6" |
| d) | agregar "2" | l) | eliminar |
| e) | agregar "8" | m) | eliminar |
| f) | eliminar | n) | agregar "4" |
| g) | agregar "1" | o) | eliminar |
| h) | agregar "9" | p) | eliminar |

11) Implemente el constructor de las clases "Pila" y "Cola", junto con los métodos "Agregar", "Quitar", "Largo" y "Vacio".

12) Suponga que quiere limitar el número de elementos de una pila, como es el caso de las pilas de recursividad que soportan hasta 1000 llamadas. ¿Qué cambios propondría para el código del punto anterior?

13) Los archivos JSON, como el archivo genome.json, se utilizan para almacenar estructuras de datos complejas, con distintos campos, que a su vez pueden estar divididos en subcampos. Estos archivos comienzan con "{" y terminan con "}". A su vez, cuando un campo posee subcampos, también se delimitan con "{" y "}". Qué estructura de datos podría utilizar para saber si un archivo está balanceado, es decir, cada campo que se abra también sea cerrado (que por cada "{" haya "}")? Implemente una función que realice este chequeo y que calcule el nivel más profundo alcanzado.

14) Un servidor de cálculo recibe procesos de todas las computadoras de un instituto de investigación. Diseñe un sistema que permita almacenar el proceso enviado, el usuario que lo envió y el número de procesadores requeridos. Cómo modificaría el sistema para admitir distintos niveles de prioridad para distintos usuarios?

15) La funcionalidad "deshacer" tiene el comportamiento de una pila, ya que deshace el último cambio registrado. Para la mayoría de los editores de texto, esta funcionalidad tiene un límite de cambios registrados y al alcanzar el límite se elimina el cambio más antiguo. Cómo implementaría una clase "Deshacer" que tenga dicho comportamiento?