



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA

Guía de Ejercicios
Estructuras lineales

Algoritmos y Estructura de datos

Curso Ing. Gustavo Schmidt

Esta es una lista de 20 ejercicios en Java, separados en 3 niveles de dificultad, utilizando vectores, listas, listas doblemente enlazadas, pilas y colas.

Nivel 1: Básico

1. **Vector de enteros:** Crear un programa que almacene 10 números enteros en un Vector, luego cargue 100, los ordene y los imprima en orden inverso.
2. **Buscar elemento en lista:** Dado una Lista (con template) de enteros, implementar un método que reciba un número y verifique si existe en la lista. Si el número está duplicado, devolver la cantidad de apariciones.
3. **Suma de elementos en un vector:** Crear un Vector con 5 números aleatorios, luego agregar 1000 números más, quitar 500, hacer que el vector reduzca su tamaño y sumar todos los elementos y mostrar el resultado de la suma y el promedio.
4. **Eliminar elemento de una lista:** Implementar un programa que permita agregar una frase a una lista (con template) de Strings, separada por palabras. Luego eliminar los elementos de una Lista que sean más cortos que una longitud dada por teclado, por ejemplo las palabras con menos de 3 letras. Para finalizar recorrer la lista con cursor e imprimir la frase resultante.
5. **Cola básica:** Crear una cola de enteros, agregar 5 elementos y luego remover los elementos, mostrando el primero en cada extracción.

Nivel 2: Intermedio

6. **Lista de palabras:** Implementar una lista de palabras utilizando Lista, y crear funciones para agregar, eliminar, y buscar palabras. Modificar el buscador y hacerlo por palabra exacto o parcialmente igual.
7. **Pila de enteros:** Crear una pila donde se puedan apilar números enteros y se imprima el tope de la pila después de cada inserción o extracción.
8. **Invertir una lista:** Implementar un método que reciba una Lista de enteros y devuelva una lista con los elementos en orden inverso, en el método utilizar una Pila.
9. **Lista doblemente enlazada:** Crear una clase Nodo para representar una lista doblemente enlazada. Implementar los métodos para agregar al inicio, agregar al final, eliminar del inicio y eliminar del final y para intercambiar una posición con otra.
10. **Conversión de decimal a binario usando una pila:** Dado un número decimal, usar una pila para convertirlo a binario y mostrar el resultado.

11. **Cola circular:** Implementar una cola circular utilizando una Cola y modificando su nodo, que permita agregar elementos en el final y eliminarlos en el inicio, manejando el desbordamiento adecuadamente.
12. **Merge de dos listas ordenadas:** Dadas dos listas enlazadas de enteros ordenadas, implementar un método que las mezcle en una sola lista también ordenada.

Nivel 3: Avanzado

13. **Pilas con múltiples operaciones:** Crear una clase que implemente una Pila (con template) de enteros con las operaciones básicas (push, pop, peek) y agregue una función getMin() que devuelva el mínimo valor actual en la pila sin eliminarlo.
14. **Cola de prioridad:** Implementar una cola de prioridad básica utilizando una lista enlazada, donde los elementos se inserten en orden según su prioridad. El primer nivel es una Lista que tiene la prioridad y el segundo nivel es una Cola con los valores.
15. **Historial de operaciones:** Simular un historial de navegación utilizando una pila. Cada vez que el usuario visita una nueva página, se agrega a la pila; el botón "atrás" saca el último sitio visitado.
16. **Lista doblemente enlazada con iteradores:** Implementar una lista doblemente enlazada que permita la creación de un segundo cursor para recorrerla en ambos sentidos a la vez.
17. **Juego de cartas con pilas:** Crear una simulación de un juego de cartas donde las cartas se manejan con una pila. Las operaciones deben incluir barajar (mezclar la pila), sacar carta (pop) y mostrar la carta superior.
18. **Tablero de juego con listas anidadas:** Implementar un tablero de juego en 2D utilizando listas de listas, donde cada celda del tablero puede almacenar un TDA Casillero. Hacer un método para obtener el casillero vecino indicando la coordenada y otro para obtener una matriz de 3x3 con todos sus vecinos.
19. **Recorrido por un laberinto:** Dado un laberinto representado por una lista de listas, usar una pila o una cola para recorrer el laberinto desde una posición inicial hasta una salida, implementando el algoritmo de búsqueda en profundidad o amplitud. El tablero es de TDA casilleros y cada uno tiene una coordenada. La pila resultante es el camino. Solo se puede mover horizontal o vertical.
20. **Simulador de colas de procesos:** Implementar un simulador que gestione varios procesos utilizando colas, donde cada proceso tiene una prioridad asignada. Las colas deben ordenarse según la prioridad de los procesos y simular el procesamiento de cada uno.