**Guía de ejercicios.**

**Temas: listas enlazadas, Pilas, Colas y Dipolos**

Lista doblemente enlazada:

1. Imprimir los elementos de una lista en orden inverso

Eliminar los últimos 3 elementos de la lista de enteros

1.5. ¿Cuál es la diferencia entre utilizar una cola, una pila o un dipolo?¿Es una lista equivalente?

2. ¿Indique al menos dos usos de las colas como estructuras de datos eficientes en una aplicación?

3. ¿Cuál sería la utilidad de una cola que contenga el método Colearse, que permite priorizar un elemento?¿Cómo sería su implementación?

4. Elabore un procedimiento que acepte como entrada una expresión aritmética simple (secuencia de números y operadores aritméticos: +, -, \* y /). Realice la evaluación de dicha expresión. Considere que la expresión aritmética viene dada en notación prefija.

Elija entre las clase Pila, Cola, Lista, Dipolo para representar la expresión e implemente las operaciones en términos de las primitivas.

Ejemplo: \*+10 2 5 → (10 + 2) \* 5 = 60.

Asuma que posee una función Operador(E) → boolean la cual retorna true si el elemento es un operador, y false en caso contrario.

5. Proponga una representación para una cola, empleando un arreglo, de manera tal que el orden de la operación encolar sea O(1), implemente el algoritmo correspondiente con el uso de su estructura.

6. Dada una cola, se desea conocer el promedio de los elementos que ella almacena. El programa debe imprimir el promedio y la cola de forma invertida.

7. Dada una cola de enteros, realice un algoritmo que muestre el elemento mayor y el menor.

8. Dada una palabra almacenada, producir otra nueva que sea el resultado de transformarla según la regla del espiral. Asuma que la estructura se encuentra almacenada en un dipolo y la estructura saliente también.

a = ( A G U A M A R I N A )

a’ = ( A A G N U I A R M A )

9. Implemente la clase cola utilizando el TDA pila y las operaciones de PILA no se puede recorrer la cola solo podemos usar las operaciones de PILA; es decir, implemente todos los métodos de una cola utilizando las operaciones de pila

10. Realice un algoritmo que dada una palabra almacenada en un dipolo, producir otra nueva en otro dipolo que sea el resultado de invertirla.

12. Dado un número decimal almacenado en una pila llevarlo a binario y dar el resultado en una cola. Ejemplo: (36)10 (100100)2.

13. Una cola medieval se comporta como una cola ordinaria, con la única diferencia de que los elementos almacenados en ella se dividen en dos grupos: nobles y plebeyos. Dentro de cada grupo, los elementos deben ser atendidos en orden de llegada; pero siempre que haya nobles en la cola, estos deben ser atendidos antes que los plebeyos. Implemente una cola medieval heredando de una cola normal, pero sobrescribiendo el método encolar. Puede realizar dos implementaciones: una de ellas simplemente re-escribiendo el método encolar, asumiendo que el tipo de dato de la cola es un boolean (que dice si es plebeyo o no). La otra solución es heredando de la clase cola original, y sobrecargando el método encolar.

14. Dado un dipolo de enteros, realice un algoritmo el cual sume los primeros N elementos desde el valor central hacia sus extremos, sin utilizar ninguna estructura de datos auxiliar. Ejemplo:

1← Tope

5

6

7

3 ←Fondo

N = 3

6 + 5 + 7 = 18

15. Dado un dipolo, realice un algoritmo que extraiga el elemento central de la misma si el dipolo posee un número impar de elementos, o los dos elementos centrales si el dipolo posee un número par de elementos. No asuma que existe una función que retorna la cantidad de elementos del dipolo ni que posee la cantidad de elementos al comenzar el algoritmo.

Ejemplo:

|  |  |
| --- | --- |
| DIPOLO 1 | DIPOLO 2 |
| 5 tope del dipolo  3  8 elemento a sacar  1  6 tope del dipolo | 8 tope del dipolo  9 elemento a sacar  2 elemento a sacar  7 tope del dipolo |

16. Suponga que tiene almacenado una secuencia de caracteres en un dipolo, y se desea saber si dicha secuencia es palíndrome.

17. Dada una cola de enteros positivos se desea obtener una pila que indique que elementos están en la cola seguido de las ocurrencias de este.

18. Dada una cola de enteros, realice un algoritmo que elimine los números negativos sin cambiar los otros elementos de las colas.

19. La notación polaca debe su nombre al famoso matemático polaco Jan Lukasiewicz, y se basa en que el símbolo operador se coloca delante de sus operandos. Asumiendo que todos los operadores a trabajar son binarios, se desea que:

• Transforme una expresión en forma normal (operando operador operando) almacenada en una lista, la almacene en una pila en notación polaca. Asuma que la expresión de entrada puede tener paréntesis.

• Elabore un algoritmo que permita evaluar una expresión en notación polaca almacenada en una pila. Ejemplo de notaciones polacas:

+ A B, que es equivalente a A + B;

+ a / b c, que es equivalente a a + b/c.

20. Utilizando únicamente las primitivas de la clase Pila, se quiere que Ud. desarrolle un procedimiento que dada una pila P, la invierta. No debe utilizar estructuras auxiliares. Luego realice el mismo algoritmo como un método de la clase pila llamado invertir, el cual sí tiene acceso a la implementación con apuntadores. Finalmente indique el orden de complejidad de cada implementación, y diga cuál es más eficiente.

21. Dada una pila de enteros, se desea que usted retorne en la misma pila, la suma de los elementos que ella almacena a pares. Ejemplo:

Al aplicar el dicho algoritmo sobre P, debe hacer que

|  |  |
| --- | --- |
| Pila | Resultado |
| 1  2  3  4  5 | 3  7  5 |

Sin emplear alguna estructura auxiliar

22. ¿Qué es una cola de prioridad y cuál es su utilidad?

23. Diseñe una estructura de datos basada en una lista enlazada desordenada cuya complejidad de encontrar el máximo valor debe ser O(1).

24. Dada una lista A (desordenada y con elementos repetidos) se quiere que usted resuelva de manera eficiente el siguiente requerimiento: Obtener una lista B que contenga los elementos que están en A sin repeticiones, seguidos por el número de ocurrencias. Resuelva el problema utilizando manejo de apuntadores. La lista A puede ser modificada.

25. Suponga un lenguaje de programación que solo permite pase de parámetros por referencia. Indique como implementaría un pase de parámetros por valor en dicho lenguaje. Justifique su respuesta.

26. Dadas dos pilas de enteros A y B ordenadas de mayor (tope) a menor (base), se desea crear una tercera pila con los elementos de ambas pilas ordenados de menor (tope) a mayor (base) sin repeticiones. Utilice la representación directa cómo un método dentro de la clase (versión 1) o utilice únicamente las operaciones de la pila (como un procedimiento fuera del alcance de la clase). Ejemplo

|  |  |  |
| --- | --- | --- |
| PILA A | PILA B | RESULTADO |
| 6 (tope)  5  3  1 | 9 (tope)  6  5 | 1 (tope)  3  5  6  9 |

27. Dada una lista de enteros, ordenada ascendentemente, escriba dos algoritmos que permita insertar en forma ordenada un entero en dicha lista.

1. Algoritmo 1: Utilizando apuntadores directamente (como método).

2. Algoritmo 2: Utilizando las operaciones de lista vistas en clase (como un algoritmo fuera del alcance de la clase).

28. Dada una palabra almacenada carácter a carácter en los nodos de un dipolo, determinar si es palíndrome. Realice este ejercicio utilizando:

a) las operaciones del dipolo

b) directamente con apuntadores, implementando palíndrome como un método

- Implementar 4 versiones dipolo:

a) Versión 1: Heredando de listas doblemente enlazadas.

b) Versión 2: Con listas doblemente enlazadas y apuntador al último.

c) Versión 3: Heredando de listas simplemente enlazadas.

d) Versión 4: Utilizando arreglos con sublista doblemente enlazada de ocupados, y sublista doblemente enlazada de libres. Cada casilla del arreglo tendrá entonces el campo “info” y dos índices (próximo y previo). La clase necesitará dos índices que apuntan al comienzo de las sublistas (priLibre y preocupado). Si alguna de las dos sublistas está vacía, el índice respectivo sería -1, o cualquier entero que esté fuera del rango de índices posibles.

29. Invertir un dipolo, utilizando métodos del dipolo, o utilizando directamente la implementación (versión 2) como un método de la clase dipolo llamado Invertir.

30. Utilizando las operaciones primitivas del TDA Pila, implante la operación mezclar, que toma dos pilas ordenadas ascendentemente y genera una tercera pila que contiene los elementos de las dos pilas dadas también ordenados en forma ascendente.

Las operaciones primitivas de Pila que podrá utilizar son:

Tope: Pila → Entero

Retorna el elemento del tope de la pila

Apilar: Pila x Entero → Pila

Retorna la pila con el nuevo elemento en el tope

Desapilar: Pila → Pila

Retorna la pila sin el tope

EsVacia: Pila → Lógico

Retorna verdadero si la pila es vacia y falso en caso contrario

CrearVacia: → Pila

Crea una nueva pila vacia

31. Implemente un procedimiento que dada una lista enlazada simple de enteros, la ordene en

forma ascendente. Considerar que pueden existir elementos repetidos.

32. Indique si las siguientes afirmaciones son ciertas o falsas. Justifique su respuesta :

a. Si un problema puede resolverse mediante procedimientos recursivos y no recursivos, el recursivo es siempre más eficiente.

b. La especificación de las primitivas definidas para un TDA cambia según la estructura (implementación) utilizada.

c. La implementación de listas mediante apuntadores es siempre más eficiente qu con arreglos (considerar espacio de memoria y tiempo de acceso).

33. Haga una corrida del siguiente programa, indicando las llamadas realizadas, el valor de las variables locales en cada ambiente de referenciación, y el valor resultante.

5 3 7 6 4 2

Función CB(N, M: entero) → entero

Si N=M o M=0 entonces

Retorna 1

Sino

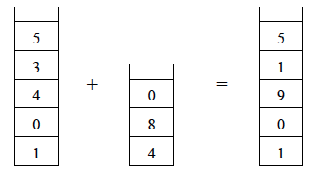
Retorna CB(N-1, M-1) + CB(N-1, M)

Fin

Fin

Llamada original: CB(4,2)

34. Dados dos enteros positivos A y B colocados dígito a dígito en sendas pilas de forma tal que el dígito menos significativo se encuentre en el tope de la correspondiente pila, se desea que usted diseñe un algoritmo recursivo que retorne otra pila con la suma de ambos enteros. Ej:



Las operaciones primitivas de Pila que podrá utilizar son:

CrearPila: → Pila

Crea una pila vacía.

EsVacía: Pila → Lógico

Función que retorna verdadero si la pila es vacía y falso en caso contrario.

Tope: Pila → Elem

Función que retorna el elemento del tope de la pila.

Desapilar: Pila → Pila

Función que retorna una pila que contiene los mismos elementos de la pila original menos

el tope.

Apilar: Elem x Pila → Pila

Función que retorna la pila con el elemento dado en el tope de la misma.

35. Una compañía que se encarga de hacer el transporte de niños de ida y vuelta entre su casa y su colegio requiere un programa que se encargue de mantener actualizadas las listas de niños que usan cada una de las 10 rutas que mantiene la compañía. Cada ruta está identificada por un número consecutivo del 1 al 10 y no existe límite predefinido de niños asignados a cada ruta (se asignarán a cada ruta tantos autobuses como sean necesarios).

Por cada niño se necesita conocer su nombre, apellido, grado, dirección y el código numérico del colegio al que va. En base a esta información se requiere que usted:

a. Plantee una estructura de datos eficiente, que permita almacenar la información requerida.

b. Implemente las siguientes operaciones de acuerdo a la estructura especificada:

i) Inscribir a un niño en una ruta determinada.

ii) Retirar a un niño de una ruta determinada.

iii) Consultar la cantidad de niños inscritos en cada una de las rutas.

36. Escriba un programa que lea una línea de la entrada y determine si la frase leída es un palíndromo. Una frase se dice que es palíndromo si se puede leer lo mismo de izquierda a derecha y de derecha a izquierda, una vez eliminados los blancos. Un ejemplo es “dábale arroz a la zorra el abad” (Recomendación: Utilice una pila y una cola)

37. En un salón de juegos de Ajedrez para niños se ha diseñado un sistema para determinar diariamente, el orden de selección de niños en grupos de a dos. Para ello los coordinadores han decidido seguir la siguiente estrategia: cada día se formará un círculo con todos los niños en orden de llegada de éstos (numerados del 1 al N, siguiendo el sentido de las agujas del reloj).

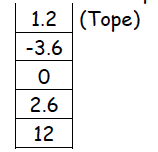
Comenzando desde 1 y moviéndose en sentido horario, uno de los coordinadores cuenta “k” niños, mientras el otro coordinador comienza en N y se mueve en sentido antihorario contando “m” niños. Estos dos niños son elegidos. Si los dos coordinadores coinciden en el mismo niño se toma el niño siguiente en el sentido horario. Cada uno de los coordinadores comienza a contar de nuevo en el siguiente niño que permanezca en el círculo y el proceso continua hasta que no quede nadie. El grupo de niños siempre es un número par y no excede de 30 niños. Se pide:

a) Diseñe las estructuras apropiadas, tomando en cuenta que el resultado final debe reflejar las “parejas” de niños en el orden en que fueron seleccionados.

b) Diseñe un algoritmo para la selección de las parejas.

38. Diseñar un algoritmo utilizando el método de intercambio que dada una lista circular que contiene números enteros los ordene en forma descendente. El ordenamiento debe hacerse sin utilizar ninguna estructura auxiliar.

39.En una pila se tienen los coeficientes de un polinomio de grado N



Esta pila representa a los coeficientes del polinomio :



40. Cree un función que evalúa, utilizando únicamente operaciones de pila, al polinomio en un

punto x cualquiera utilizando la Regla de Horner:



Operaciones válidas:

Tope( Pila A), Desapilar (Pila A), Vacío (Pila A), Apilar( Pila A, Número N)

Restricciones: La pila debe quedar intacta al final y no se puede utilizar una pila auxiliar.

41.Dada una lista enlazada de enteros. Realice el procedimiento recursivo que determina la

longitud de la secuencia creciente más larga que contiene. Por ejemplo:

La lista: 3 5 11 2 9 15 10 9 9 19 22 24 18

En este caso la respuesta es 5 ( secuencia: 9 9 19 22 24)

42.Un dipolo es una estructura de lista que tiene como indicador una variable estructurada

que contiene dos punteros simples a casillas de lista, uno apunta a la cabeza de la lista y el

otro a la cola de la misma :

Struct dipolo {

Nodo \*cabeza;

Nodo \*cola }

Implemente las siguientes primitivas del TDA lista utilizando dipolos.

- Insertar (Dipolo A, Entero N) (Inserta ordenadamente el entero N en la lista A)

- Eliminar (Dipolo A, Entero N) (Elimina el entero N, de la lista A)

43. Muestre el árbol de ejecución del algoritmo siguiente:

Función RC(r: entero, x: entero) → entero

si (r = -1) entonces

retornar –1;

sino

si (M[r].A = x) entonces

retornar r;

sino

si (M[r].A > x) entonces

retornar RC(M[r].B, x);

sino

retornar RC(M[r].C, x);

fsi

fsi

fsi

función

Tipo T = registro

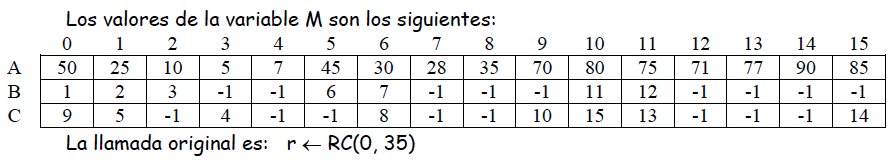
A : entero;

B : entero;

C : entero;

fregistro

Var M: arreglo [0..15] de T;



44. En una cárcel de alta seguridad, se encuentran los prisioneros más peligrosos que han sido condenados a muerte. Existe en la penitenciaría, un sistema automatizado que lleva registrados los condenados a muerte en una “lista”, donde el primer elemento es el preso más antiguo y éste tiene el número uno (1). Cuando se activa la pena de muerte en el país, se activa un proceso de “selección” que escoge de la lista los presos que se ejecutarán, como la ley humana no es infalible quieren dejar algo para la ley de Dios, Así que este proceso se conoce como “SUPERVIVIENTES” ya que comienza seleccionando un preso de cada “n” (, por ejemplo, si n = 2 el preso 1, 3, 5,…..), luego, con la lista resultante escoge uno de cada “n+1”, luego, uno de cada “n+2”, y así sucesivamente. El proceso termina cuando se va a eliminar un número de cada “m” y el tamaño de la lista es menor a “m”. Los presos que se queden en la cola en ese momento son los supervivientes. Se pide diseñar “SUPERVIVIENTES” recursivo que reciba la lista de los presos y un n inicial. El proceso debe devolver a los sobrevivientes en una cola de manera que el frente de la misma contenga el primer afortunado.

Las operaciones primitivas de que podrá utilizar para las colas son: Frente, Encolar, Desencolar, Colavacía y

CrearCola

45. Una expresión aritmética construida con los operadores aritméticos binarios ‘+’,’ –‘, ‘\*’ , ‘/’ y operandos de un único dígito entre 0 y 9, se dice que está en forma pos-fija cuando el operador binario está al final de los dos operandos. Los operandos a su vez que lo conforman pueden ser a su vez, expresiones en forma post-fija, aplicando el mismo concepto. Se asumirá que la expresión está totalmente agrupada entre paréntesis para no tener problema con la escogencia de prioridad de los operadores. A continuación se muestra un ejemplo:

La expresión aritmética:

(((6 + 4) / 5) + (8 \* (3/(5\*2)))) (forma in-fija)

Tiene la forma post-fija: 6 4 + 5 / 8 3 5 2 \* / \* +

Se asume que la expresión está en una pila donde cada nodo contiene un elemento que puede ser un paréntesis, un número o un operador aritmético. En el tope de la pila está el elemento más a la izquierda de la expresión (en el ejemplo, el tope sería “(“). Se asumirá también que la expresión está bien construida. Se pide:

Diseñe un algoritmo que dada la pila que contiene la expresión en forma in-fija, construya la expresión en forma post-fija y la deje en otra pila (donde el tope será también el elemento más a la izquierda.

Puede usar estructuras auxiliares (dinámicas)

46. Una cola se representa con la siguiente estructura de datos

struct tcola {

int final, frente;

int elementos [50]

}

Si no se guarda el número de elementos de la cola, si frente = 38 y final = 36, ¿Cómo está la cola?

i) La cola está completamente llena

ii) La cola está vacía.

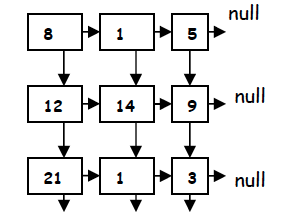
iii) La cola está casi llena excepto un elemento.

iv) La cola está casi vacía

v) Ninguna de las anteriores

47. Una matriz puede representarse como una estructura dinámica enlazando filas y columnas

de la siguiente forma:



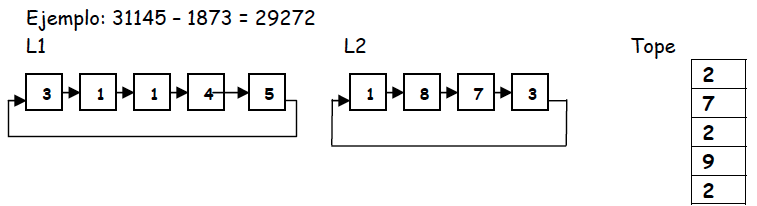
Se define el punto de ensilladura como la posición i,j que contiene al mínimo valor de la fila i que corresponde al máximo valor de la columna j.

En el ejemplo, la matriz posee un punto de ensilladura en la posición 2,3 (mínimo valor de la fila 2 que corresponde al máximo valor de la columna 3). Escriba una función que escriba los valores de i y de j del punto de ensilladura de una matriz dada en el caso de existir. En caso contrario debe devolver los valores -1,-1.

48. Dos números enteros se encuentran representados en dos listas circulares simplemente enlazadas, de forma tal que cada nodo contiene un dígito entre 0 y 9. El dígito más significativo se encuentra en la primera posición de la lista.

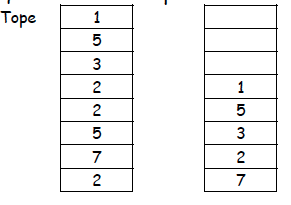
Se pide implementar una función RESTA, que reciba como parámetro las dos listas y devuelva como resultado una pila con la resta de ambos números. En el tope de la pila quedará el dígito menos significativo.

Ejemplo: 31145 – 1873 = 29272



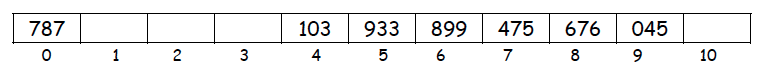
No se permite utilizar estructuras auxiliares y la pila debe trabajarse solo con las primitivas vistas en clase.

49. Dada una pila que contiene enteros, se pide desarrollar un procedimiento recursivo que elimine todos los elementos duplicados. Ejemplo: la pila con los elementos: 1, 5, 3, 2, 2, 5, 7, 2 quedaría tras la aplicación de la operación como: 1, 5, 3,2, 7.



Use solamente las primitivas vistas en clase. No puede utilizar estructuras auxiliares.

50. Se aplicó la función Hashing: H(clave) = ( Σ di ) MOD 11 y se utilizó con estrategia de manejos de colisiones el método la prueba lineal o secuencial. Sobre la tabla de 11 posiciones se insertaron siete claves, como se muestra a continuación:



Indique cuál de las siguientes afirmaciones es falsa:

i) La clave 899 colisionó al momento de la inserción

ii) Un posible orden en que las claves se insertaron es: 103, 475, 787, 045, 933, 676, 899

iii) La inserción de la clave 676 no provocó colisión alguna

iv) La búsqueda de la clave 103 requiere sólo una comparación

v) Ninguno de los anteriores.