



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:

M.I.Marco Antonio Martínez Quintana

Asignatura:

Estructura de Datos y Algoritmos I

Grupo:

15

No de Práctica(s):

6

Integrante(s):

Matias Zavala Melissa Maruuati

*No. de Equipo de
cómputo empleado:*

No aplica

No. de Lista o Brigada:

No aplica

Semestre:

2021-2

Fecha de entrega:

9 de agosto 2021

Observaciones:

CALIFICACIÓN: _____

Objetivo

Revisar definiciones, características, procedimientos y ejemplos de las estructuras lineales Pila, Cola, Lista simple y Lista circular, para comprender sus estructuras e implementarlas.

Introducción

La estructura de datos consiste en una colección de registros del mismo tipo, los cuales se relacionan entre sí. También se les puede llamar nodos, la cual es la unidad mínima de almacenamiento de información en la estructura de datos.

Las estructuras de datos lineales son aquellas en la que cada elemento ocupa un lugar seguido en la estructura y cada uno tiene un único sucesor, así como un único predecesor.

Las listas son un tipo de estructura de datos lineal y dinámica, dinámica porque su tamaño no es fijo y se puede definir conforme se requiera. Las operaciones básicas dentro de la lista son BUSCAR, INSERTAR y ELIMINAR.

Pila

Estructura de datos lineal y dinámica también llamada *stack*, donde cada elemento obtenido por medio de la operación ELIMINAR está predefinido, debido a la política *Last-In, First-Out (LIFO)*, es decir, el último elemento agregado es el primero en eliminar.

Sobre esta se pueden realizar las operaciones:

- INSERTAR (*push*)
- ELIMINAR (*pop*)

El elemento accesible que se encuentra hasta arriba se conoce como tope de la pila.

Al diseñar un algoritmo con el comportamiento de una pila, es necesario considerar 3 casos para las operaciones ya mencionadas:

- Estructura vacía (caso extremo)
- Estructura llena (caso extremo)
- Estructura con elemento(s) (caso base)

Pila vacía

No contiene elemento alguno dentro de la estructura y el tope es nulo.

Para esta pila no es posible realizar POP, ya que no contiene información, pero si se puede realizar PUSH, siendo el nodo que está, el único elemento de la lista y el tope apuntaría a él.

Pila llena

La pila tiene un tamaño fijo, por lo tanto, cuando la pila ha almacenado el número máximo de nodos definido está llena.

Caso contrario que en la vacía, ya no le es posible hacer PUSH de un nuevo elemento, siendo que ha alcanzado el tamaño máximo permitido. Para este caso se puede hacer POP de la

información contenido, donde el tope apunta al elemento siguiente de la estructura.

Pila con elementos

Una pila que contiene elementos (sin llegar a la máxima capacidad) representa el caso general. En esta se pueden realizar tanto PUSH como POP, en el primero el tope apuntara al elemento insertado y el nuevo elemento apunta al que apuntaba el tope; para el segundo, el nodo al que apunta el tope se extrae y entonces el tope apunta al sucesor.

Cola

Estructura de datos lineal, donde el elemento obtenido a través de la operación ELIMINAR está predefinido y se encuentra al principio de la estructura. A diferencia de la pila, la cola implementa la política *First-In, First-Out (FIFO)*, esto es, el primer elemento agregado, es el primero que se elimina.

Es de tamaño fijo y sus operaciones se realizan por ambos extremos, permite INSERTAR (ENCOLAR) elementos al final de la estructura y ELIMINAR (DESENCOLAR) elementos por el inicio de la misma.

Al diseñar un algoritmo con el comportamiento de una pila, es necesario considerar 3 casos para las operaciones ya mencionadas:

- Estructura vacía (caso extremo)
- Estructura llena (caso extremo)
- Estructura con elemento(s) (caso base)

Cola vacía

La cola tiene dos referencias, al inicio (HEAD) y al final (TAIL) de la cola. En la vacía, ambas referencias apuntan a nulo. No es posible desencolar ya que no posee elementos, pero sí se puede encolar elementos, donde las referencias apuntan al mismo elemento (único de la estructura).

Cola llena

Al llegar la referencia tail a su máxima capacidad de almacenamiento (MAX), la cola está llena. Donde no es posible encolar más elementos, aunque si desencolarlos y se obtiene al elemento que hace referencia a head y elemento que se recorre al sucesor.

Cola con elementos

Aquella que contiene elementos sin llegar a su tamaño máximo, representa el caso general de la estructura. Es posible desencolar nodos, recorriendo la referencia al inicio de la cola (HEAD) al siguiente elemento de la estructura. También se pueden encolar elementos mientras la referencia al final (TAIL), no sea mayor al tamaño máximo de la misma. Cuando se encola un elemento, el nodo que apunta a TAIL, tiene como sucesor el nuevo elemento y la referencia TAIL igual.

Lista simple

Conocida igualmente como lista ligada o lista simplemente ligada, está constituida por un conjunto de nodos alineados de manera lineal y unidos entre sí por una referencia. A diferencia del arreglo, el orden está determinado por una referencia, no por un índice, y el tamaño no es fijo.

La unidad básica de la lista simple es un elemento o nodo. Cada elemento es un objeto que contiene la información a almacenar, así como una referencia (NEXT) al siguiente elemento (SUCESOR).

Para poder diseñar un algoritmo que defina el comportamiento de una LISTA LIGADA, se deben considerar 2 casos para cada operación (BUSCAR, INSERTAR y ELIMINAR):

- Estructura vacía (casi extremo)
- Estructura con elemento(s)(caso base)

Buscar

El método debe buscar el primer elemento que coincida con la llave K dentro de la lista L , a través de una búsqueda lineal simple, regresando un apuntador a dicho elemento si éste se encuentra en la lista o nulo en caso contrario.

Una lista simple vacía no contiene elementos, la referencia al inicio de la misma (head) apunta a nulo, por lo que no es posible buscar elementos. Mientras tanto, en una lista simple con elementos, puede contener de 1 a n elementos, la referencia al inicio (HEAD) apunta al primer elemento de la lista. Es posible recorrer la lista a través de la referencia (NEXT) de cada nodo hasta llegar al que apunta nulo (último elemento), por lo que es posible buscar una llave K .

Inserta

Dado un nodo x que contenga una llave k , previamente establecida, el método INSERTAR agrega el elemento x al inicio de la lista.

Es posible insertar elementos tanto en una lista simple vacía como en una con elementos. Cuando se inserta un nuevo elemento, la referencia al inicio de la lista (HEAD) apunta al nodo insertado. Al insertar un nuevo elemento en una lista simple con elementos, la referencia del nuevo nodo (NEXT) apunta al mismo nodo al que apunta el inicio de la lista (HEAD) y HEAD apunta al nuevo nodo.

Borrar

El método elimina el elemento x de la lista L (si se encuentra en la estructura). Para eliminar un elemento de la lista primero es necesario saber la ubicación del nodo a eliminar, por lo tanto, primero se realiza una búsqueda del elemento.

En una lista simple vacía no es posible eliminar, ya que no contiene elementos. Para eliminar un nodo en una lista simple con elementos, primero se busca el elemento a eliminar, una vez encontrado del nodo en la lista, se mueven las referencias de la estructura de tal manera que el antecesor del nodo a eliminar apunte al sucesor del mismo.

Lista circular

Una lista simplemente ligada modificada, donde el apuntador del elemento encontrado al final de la lista (TAIL) apunta al primer elemento de la lista (HEAD).

Buscar

El método debe buscar el primer elemento que coincida con la llave K dentro de la lista L, a través de una búsqueda lineal simple, regresando un apuntador al elemento si éste se encuentra en la lista o nulo en caso contrario.

Una lista circular vacía no contiene elementos, la referencia al inicio de la misma (HEAD) apunta a nulo, por lo tanto, no es posible buscar elementos.

Una con elementos puede contener de 1 a n elemento, por lo cual la referencia al inicio (HEAD) apunta al primer elemento de la lista y la referencia a NEXT del último elemento apunta al primer elemento. Es posible recorrer la lista a través de la referencia (NEXT) de cada nodo, se debe tener el número de elementos de la lista, ya que el último apunta al inicio de la estructura, por ello se puede recorrer de manera infinita. Dentro de esta es posible buscar una llave K.

Insertar

Dado un nodo x que contenga una llave K previamente establecida, este agrega el elemento x al inicio de la lista.

Se pueden insertar elementos tanto en una lista circular vacía como en una con elementos. Cuando se inserta un nuevo elemento en una lista circular vacía la referencia al inicio de la lista (HEAD) apunta al nodo insertado y la referencia a NEXT del nodo apunta a sí mismo. Al insertar un nuevo elemento en una lista circular con elementos, la referencia del nuevo nodo (NEXT) apunta al mismo nodo al que apunta el inicio de la lista (HEAD) y ahora este apunta al nuevo nodo, el último nodo de la estructura (TAIL) apunta al primer elemento.

Borrar

El método elimina el elemento x de la lista L (si está en la estructura). En la lista circular vacía no es posible eliminar, ya que no contiene elementos. Para eliminar un elemento, primero se debe buscar el elemento a eliminar, una vez encontrado el nodo de la lista, se mueven las referencias de la estructura de tal manera que el antecesor del nodo a eliminar apunte al sucesor del mismo.

Desarrollo

Aplicación de colas

1. En el momento de enviar una serie de impresiones, de cómo va de una en uno, siendo el primero a imprimir, eliminando y sigue el sucesor del elemento.
2. Para quienes atienden los teléfonos, son una serie de personas, entonces el primero que llamo es atendido y después sigue el otro para atender

3. Plan de actividades, donde se hace una serie de cosas a hacer, se hace la primera y se sigue con la otra.

Aplicación de listas

1. Facebook (red social), donde se hace una lista de cada publicación sucesora, donde cada una tiene un único elemento sucesor, la cual sería la publicación anterior.
2. Spotify (reproducción de música), aquí con la reproducción aleatoria o reproducir en orden, donde se cumple la parte lineal y tiene una extensión dinámica.
3. Ordenar el registro de una base de datos, donde se pueden ir añadiendo más dato y organizando.

Conclusiones

Entendí cada una de las estructuras mencionadas (pilas, colas, lista simple y lista circular, junto con las características que tienen, los procedimientos que sigue cada una para trabajar y más que nada como se puede implementar cada una de ellas. También se me hace compleja la parte de lista circular.

Referencias:

Manual de prácticas del laboratorio de Estructuras de datos y algoritmos I. Facultad de Ingeniería UNAM. Recuperado de: <http://lcp02.fi-b.unam.mx/>