

GUÍA DE TRABAJOS PRÁCTICOS

CURSADA 2024 – 1ER CUATRIMESTRE

Práctica III: Pilas

En cada caso se debe implementar el TAD descrito y una aplicación que, utilizando el tipo abstracto de dato, permita al usuario el ingreso de una o más instancias (según sea necesario), la aplicación de cualquiera de las operaciones y luego la obtención del resultado.

1. Construir el TAD de pilas correspondientes a las implementaciones de “Arreglos” y “Punteros” teniendo en cuenta:
 - a. Definir las estructuras de datos en función a las especificaciones dadas en clase y documentadas en los PDF que se encuentran en el aula virtual de la asignatura.
 - b. Se las debe implementar y respetar estrictamente los nombres y parámetros de cada una de las operaciones del TAD especificadas.
 - c. Se las debe probar y testear de forma tal que se pueda asegurar que la TAD funciona correctamente.

Importante !!!

Los siguientes ejercicios siguientes deben se implementados y resueltos en forma genérica, esto significa que se debería poder referenciar cualquiera de las implementaciones de pilas (arreglo y Puntero) y los mismos deben seguir en funcionamiento sin problemas.

2. Dada una pila cargada con valores al azar realizar los siguientes ejercicios:
 - a. Buscar una clave y determinar si existe en la Pila (sin perder la pila).
 - b. Colocar en una posición ordinal determinada, recibida por parámetro, un nuevo elemento (Insertar un elemento nuevo).
 - c. Eliminar de una pila un elemento dado (primera ocurrencia encontrada por la clave).
 - d. Intercambiar los valores de 2 posiciones ordinales de la pila, por ejemplo la 2da con la 4ta.
 - e. Duplicar el contenido de una pila.
 - f. Contar los elementos de la pila.
3. Dadas dos pilas, determinar si sus contenidos son exactamente iguales (solo por la clave), sin destruirlas. Utilizar para la resolución del problema una única pila auxiliar. Determinar la complejidad algorítmica de la solución.
4. Construir un algoritmo que, utilizando el TAD Pila, permita convertir un número decimal pasado como parámetro, a su correspondiente valor expresado en una base de 2 a 16 (hexadecimal). Determinar la complejidad algorítmica de la solución.
5. Invertir el contenido de una pila sin destruir la pila original. Se debe retornar la nueva pila invertida y la original no se debe perder. Determinar la complejidad algorítmica de la solución.

6. Dada una pila con valores al azar eliminar todas las ocurrencias de un determinado ítem sin perder la pila original. Deberá retornar una nueva pila sin el ítem en consideración. Resolver **iterativamente y recursivamente**. Determinar la complejidad algorítmica de ambas soluciones.

Ejemplo: si “P” contiene (1, 5, 7, 1, 3, 1, 8) y el ítem a eliminar es “1” entonces la pila resultante sería (5, 7, 3, 8).

7. Dada dos pilas realizar una función que permita determinar los elementos que tienen en común. Debe retornar una nueva pila con esos elementos (que están en ambas pilas) sin perder las pilas originales. Determinar la complejidad algorítmica de la solución.

Ejemplo: si “P1” contiene (1, 4, 7, 9, 8) y “P2” contiene (4,1,6,9,3) el resultado será la pila “PR” = (1,4,9).

8. Dada una pila con valores repetidos, se desea obtener una nueva pila con todos los valores (sin repetición) y la cantidad de veces que aparecen. No debe perderse la pila original. El proceso recibe la pila con las repeticiones y retorna una nueva pila. Determinar la complejidad algorítmica de la solución.

Ejemplo: si “P” contiene (1, 2,1, 3, 4, 5, 2, 1, 3, 5, 5), la pila resultado del proceso será “PR” = (1:3, 2:2, 3:2, 4:1, 5:3).

IMPORTANTE !!!

- Los prácticos podrán ser resueltos en grupos de 4 **(tres) a 7 (seis)** alumnos. Condición indispensable para que los TPs sean evaluados.
- Todos los prácticos son obligatorios.
- Los prácticos deben ser entregados en tiempo y forma en las fechas que los docentes estipulen (serán determinadas según la cursada y avisadas mediante la plataforma virtual).
- Los prácticos NO se recuperan por lo que si son entregados en condiciones incorrectas de funcionamiento, fuera de término o fuera de formalidad (abajo descripto) se consideran desaprobados.
- Se deben presentar todos los ejercicios del tp. Se deben tener como mínimo un 60% de los ejercicios correctos para aprobar la práctica, con 2 de los 3 últimos ejercicios funcionando correctamente. Se evalúan tanto los errores de lógica como los de ingreso de datos. Deben controlar los ingresos de datos, tales como blancos, ingreso de números correctos, etc.
- Los prácticos deben presentarse con la formalidad:
 - Carátula Impresa o PDF por mail con: Fecha de Entrega, Integrantes de Grupo, Título del Práctico a Entregar y Nro de Grupo si lo tienen.
 - Se tomará asistencia en el momento de entrega del TP si es presencial.



- El código fuente debe estar disponible en GIT HUB, con acceso a los docentes.
- Si se entregaron los prototipos (.h) de los ejercicios se deben respetar.
- Los ejercicios deben ser posible probarlos en cualquiera de las implementaciones de la pila (arreglos o apuntadores).

El lenguaje de programación será “C”. (Usar el foro para consultas sobre el TP)

FECHA DE ENTREGA: 22/04/2024.

Profesor a Cargo de la Asignatura: Mario Perello.

Trabajos Prácticos a Cargo de: Claudia Reinaudi, José Racker, Pablo Chale y Mariano Goldman.