

PRACTICA # 5: ALGORITMOS DE BÚSQUEDA PARTE 2

OBJETIVO: el estudiante conocerá e identificará las características necesarias para realizar búsquedas por transformación de llaves

Objetivo de clase: El alumno conocerá la implementación de la transformación de llaves en el lenguaje orientado a objetos

ACTIVIDADES PARA EL DESARROLLO DE LA PRÁCTICA**Ejercicio 0. Menú para la práctica**

- a) Con ayuda de NetBeans, crea un nuevo proyecto llamado Practica5[ApPaternoNombre].
- b) Crea un menú en el cual puedas seleccionar entre las opciones siguientes
 - 1) Manejo de Tablas Hash en Java
 - 2) Función hash por módulo
 - 3) Encadenamiento

Ejercicio 1. Tablas Hash en Java

El manejo de tablas hash en el lenguaje de programación Java, se realiza a través de la interfaz Map<E>, dicha interfaz cuenta con varias implementaciones. Las más utilizadas son HashMap y TreeMap, la diferencia fundamental entre ambas es que la primera permite elementos nulos, conserva un cierto orden en su estructura y no es “sincronizada”; en tanto que la segunda no permite elementos nulos, los elementos no tienen un orden específico y maneja la sincronización.

Al igual que la práctica anterior, deberás crear una TablaHash en java (puede ser HashMap HashTable), para la tabla hash deberás crear parejas donde el valor será un “String” que representa el nombre (completo) de un alumno y la clave de la tabla hash será un numero entero que representa el número de cuenta.

Agrega algunos elementos a la tabla hash. Y posteriormente, con la tabla creada prueba los siguientes métodos:

- Contains
- containsKey
- containsValue
- equals
- get
- put
- remove
- size

Si lo deseas puedes agregar otros métodos de tablas hash y explicar su funcionamiento

Ejercicio 2. Simulación de Función Hash por módulo.

- a) Agrega una clase al proyecto llamada HashMódulo, en ella, crea una nueva lista (puede ser linked list o array list) de enteros; el tamaño de la lista se definirá desde un principio. Para efectos de la práctica será de 15 (para crear una lista de 15 elementos debes inicializar todos sus elementos con null)

CONTINÚA EN LA PARTE POSTERIOR...

Elabora un submenú para que el usuario pueda seleccionar entre las siguientes opciones

- 1) Agregar elementos
- 2) Imprimir lista
- 3) Buscar elementos

b) Crea un método donde se utilizará la función hash por módulo cada vez que el usuario agregue o busque elementos en la lista. Considera lo siguiente:

- ✓ El valor de M para la función hash por módulo será un valor $M=13$.
- ✓ Cada vez que el usuario agregue un nuevo valor, la función hash calculará el índice que ocupará en la lista ese valor.
- ✓ En caso de haber colisiones, se resolverán por el método de PRUEBA LINEAL

c) Al agregar o buscar elementos el programa deberá informar al usuario el resultado de la función hash aplicada indicando la posición donde se almacenó/encontró el elemento.

Ejercicio 3. Encadenamiento

Agrega una nueva clase al proyecto llamada *Encadenamiento*, en ella deberás realizar lo siguiente:

a) Deberás crear una lista de 15 listas de enteros.

b) Implementa una función aleatoria que determinará la posición de los elementos ingresados por el usuario.

El usuario podrá elegir, agregar elemento o salir de la sección

Cada vez que el usuario ingrese un número, la función aleatoria indicará la posición donde se ubicará el número y mostrará el estado global de la lista.

Elabora las conclusiones de tu práctica indicando si se lograron los objetivos de aprendizaje.