



TDA Hash

[7541/9515] Algoritmos y Programación II
Primer cuatrimestre de 2022

Alumno:	de San Vicente, Hernán
Número de padrón:	108800
Email:	hdesanvicente@fi.uba.ar

Índice

1. Introducción	2
2. Teoría	2
2.1. Tabla de Hash abierta	2
2.2. Tabla de Hash cerrada	2
3. Detalles de implementación	2
3.1. Estructura del hash	2
3.2. Función de Hashing	2
3.3. Función y criterio de Rehashing	2
3.4. Otras funciones	3
4. Diagramas	3

1. Introducción

En este trabajo se nos pidió implementar las funciones primitivas principales de un TDA Hash. Se nos pidió que éste sea un hash abierto con direccionamiento cerrado.

2. Teoría

Además de programar el TDA, se nos pidió responder una serie de preguntas y explicar ciertos conceptos. Esto es explicar qué es una tabla de hash abierta y cerrada e ilustrar las operaciones de inserción, eliminación y búsqueda para los dos casos.

2.1. Tabla de Hash abierta

Explicación de qué es. Operaciones:

- Inserción:
- Eliminación:
- Búsqueda:

2.2. Tabla de Hash cerrada

Explicación de qué es. Operaciones:

- Inserción:
- Eliminación:
- Búsqueda:

3. Detalles de implementación

En esta sección se van a explicar la implementación y decisiones de diseño.

3.1. Estructura del hash

En este trabajo se implementó una estructura de pares que contienen su respectiva clave y valor (el elemento). Estos pares son guardados en listas de nodos enlazados cuyas referencias son guardadas en un vector, la tabla de hash. Este vector de punteros a listas se encuentra en otra estructura, la del hash, que además tiene la capacidad y la cantidad de datos guardados.

3.2. Función de Hashing

La función de hashing fue tomada de internet y es utilizada por ser muy eficiente. No se utiliza el valor en sí de esta función para agregar pares a la tabla, sino que se utiliza el resultado del módulo de ésta y la capacidad de la tabla.

3.3. Función y criterio de Rehashing

Rehashear una tabla de hash implica volver a insertar los elementos de una tabla de hash para que estén mejor distribuidos. El criterio para decidir cuándo realizar esta costosa operación depende del porcentaje de espacio usado de la tabla de hash. Si el espacio usado (la cantidad) excede el 75 % de la capacidad, se rehashea.

3.4. Otras funciones

1. Creación y destrucción: Las funciones de creación simplemente reservan memoria para las diferentes estructuras (hash, lista y par) y las de destrucción liberan esa memoria. La función destruir todo, particularmente, además aplica cierta función pasada por parámetro a cada uno de los elementos de la tabla.
2. Insertar, obtener y quitar: La función de inserción determina la posición del elemento a partir de su clave, utilizando la función hash. En caso de no existir esa clave, se inserta al final de la lista en su respectiva posición, si existe se reemplaza el elemento. Obtener simplemente busca la clave y devuelve el elemento. La función de quitar busca la clave dada y destruye el par, si es que se encuentra, devolviendo el elemento destruido.
3. Contiene y cantidad: Contiene simplemente busca el elemento y devuelve true o false si la búsqueda es exitosa o no. Cantidad devuelve el valor almacenado en la estructura de hash.
4. Con cada clave: Esta función empieza a recorrer elementos de la tabla aplicando cierta función pasada por parámetro, hasta que se termine la tabla o la función devuelva false, devolviendo la cantidad de elementos recorridos.

4. Diagramas

En esta sección van los diagramas que realizas para poder acompañar los detalles de implementación y de funciones que escribiste mas arriba. Trata de que sean lo mas claros posibles, puedes hacer mas de un diagrama para una sola función, por ejemplo puedes tener un diagrama por cada paso que realiza la función en vez de tener un solo diagrama (que a lo mejor termina siendo poco claro) con toda la información metida junta.

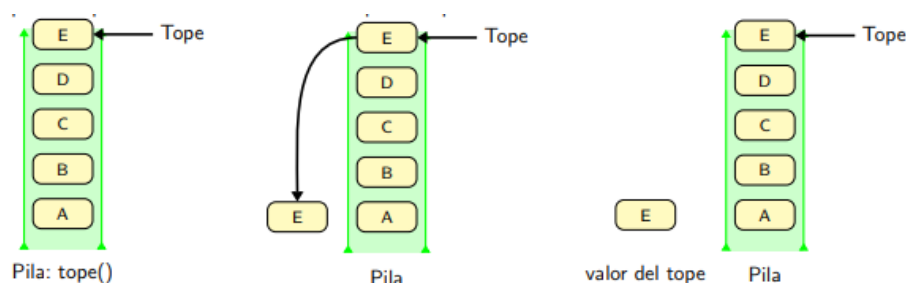


Figura 1: Ejemplo de un diagrama.

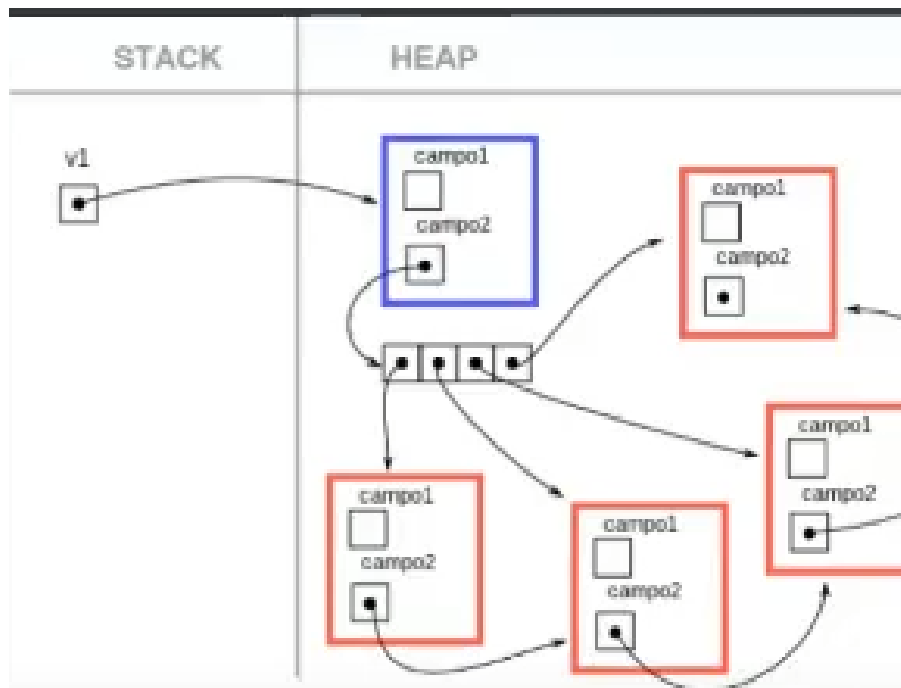


Figura 2: Otro ejemplo de un diagrama.