DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 2

Diseño e implementación de estructuras

23 de junio de 2024 — Algoritmos y Estructuras de Datos

GRUPO EPICO - LSSPMSSLDBLMOLTIKQGP

Integrante	LU	Correo electrónico
Condori, Alex	163/23	nocwe11@gmail.com
Della Rosa, Facundo	1317/23	dellarosafacundo@gmail.com
García, Lucía	706/22	luciiamar19@gmail.com
Ser, Gonzalo Nicolás	573/21	ser.gonzalo10@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

1. Invariantes de Representación de las clases de Java

1.1. Trie

- Todos los nodos tienen un array -hijos- de 256 posiciones que representan el código ASCII de cada caracter.
- Los atributos palabra, padre y valor del nodo raíz son Null.
- Agregar una clave que ya está en el Trie pisará su valor.
- Los hijos de un nodo no pueden tener caminos que apunten a distinto nodo pero de igual carácter. Es decir, cada hijo conduce a una única palabra.
- Los nodos que no son la raíz tienen un puntero a su nodo padre.
- El tamaño del Trie siempre es igual a la cantidad de claves.
- Intentar obtener una clave que no esté en el Trie retornará Null.
- Cada clave está formada por el recorrido de los nodos correspondientes a sus caracteres.
- Para cada nodo, su array de hijos tienen valores nulos en las posiciones que no llevan a una clave.
- En el último nodo del recorrido de la clave se encuentra el valor y un String -palabra- correspondiente a dicha clave.
- Cada nodo es alcanzable, es decir que se puede acceder a todos los nodos que estén en el Trie.

1.2. DatosMateria

- La cantidad de estudiantes es siempre ≥ 0.
- La longitud de la lista de libretas es igual a la cantidad de estudiantes.
- La longitud de cada elemento de la lista de libretas tiene como máximo 7 caracteres (xxxx/xx).
- La longitud del Array “Docentes por cargo” es 4.
- Toda posición en el Array “Docentes por cargo” es un entero positivo.
- El cupo disponible es igual al mínimo entre: 250 * cantidad de Profesores, 100 * cantidad de JTPs, 20 * cantidad de Ayudantes de 1° y 30 * cantidad de Ayudantes de 2° menos la cantidad de estudiantes.
- Cada elemento de la lista “Nombres materia” contiene un puntero al Trie de materias de una carrera y el nombre que tiene la materia en dicha carrera.
- Cumple el Invariante de Representación de **ParRefCarreraMateria.**

1.3. ParRefCarreraMateria

- Ref Carrera contiene una referencia a un Trie de materias de una carrera que pertenece al Trie Carreras del Sistema SIU.
- Materia es el nombre que tiene una materia de la carrera referenciada por RefCarrera. La materia pertenece al Trie de materias de dicha carrera.

1.4. ListaEnlazada

- Todos los nodos, exceptuando al Primero, son accesibles a través de su anterior.
- Para 2 nodos a y b cualesquiera, si el siguiente de a es b, entonces el anterior de b es a.
- El atributo Size es igual la cantidad de nodos.
- El puntero siguiente al último nodo apunta a null.
- El puntero anterior al primer nodo apunta a null.
- Cada nodo tiene 2 punteros que apuntan hacia él.

1.5. SistemaSiu

- el Trie Estudiantes contiene las libretas universitarias de todos los estudiantes inscriptos en el sistema. El valor de sus elementos es la cantidad de materias en las que está inscripto el estudiante con esa libreta.
- Cada valor del trie "estudiantes" tiene como máximo 7 caracteres (xxxx/xx).
- El Trie Carreras contiene los nombres de todas las carreras que están en el sistema. El valor de sus elementos es una referencia al Trie de materias.
- El Trie de materias contiene todas las materias de una carrera que están en el sistema. El valor de sus elementos es del tipo DatosMateria.
- Cada Trie cumple la invariante de representación del Trie escrito anteriormente.
- Cada ListaEnlazada cumple la invariante de representación de listaEnlazada escrito anteriormente.
- Cada DatosMateria cumple la invariante de representación de DatosMateria escrito anteriormente.

2. Aclaraciones

2.1. complejidad temporal del constructor del SistemaSIU

El constructor del SistemSIU va recorriendo las carreras y materias en un orden distinto al que está en el enunciado. El algoritmo va recorriendo el Array de infoMaterias (el primer for loop). Cada elemento de este Array contiene a su vez un Array con paresCarreraMateria (el segundo for loop). Cada parCarreraMateria tiene un String materia y un String carrera. Primero, o bien recorremos o bien creamos y agregamos la carrera al Trie Carreras, según corresponda. En total esto se realiza tantas veces como materias agreguemos a esa carrera.

$$O(\sum_{c \in C} \mid c \mid * \mid M_c \mid) \tag{1}$$

Segundo, dentro del Trie de materias de esa carrera, agregamos la materia del par. Esta operación en total la realizamos tantas veces como materias vayamos a agregar (contando todos sus nombres).

$$O(\sum_{m \in M} \sum_{n \in N_m} \mid n \mid) \tag{2}$$

Por último, simplemente recorremos la lista de Estudiantes y agregamos cada String a una lista enlazada.

$$O(E) \tag{3}$$

En conjunto, las complejidades se suman y cumplen con lo pedido en el enunciado.