

Organización y Manejo de Archivos

Tarea #1

Hashing lineal

El objetivo de la presente tarea es la implementación de un algoritmo de indexación basado en hashing lineal en memoria principal. El algoritmo de inserción es el sgte:

- se inserta un registro con clave K sólo cuando se ha calculado el valor de la función de transformación H .
- se toman los últimos d dígitos de H .
- buscar el bucket m donde K debe ser almacenada.
- si $(m < n) \rightarrow$ existe un bucket para almacenar K , si el bucket no tiene espacio usar bucket de overflow.
- si $(m \geq n) \rightarrow$ almacenar K en el bucket $m - 2^{d-1}$.
- luego de cada inserción chequear si $fc < umbral$.
- si $(fc > umbral) \rightarrow$ incrementar el espacio de almacenamiento (split) considerando:
 - agregar un nuevo bucket (esto puede incrementar d).
 - redistribuir los registros entre el nuevo bucket $n = 1b_2...b_d$ y el bucket $n = 0b_2...b_d$.

Datos de entrada

Los datos a indexar se encuentran en un archivo llamado **datos.txt** y corresponde a un archivo de texto, cuyos registros poseen el siguiente formato:

- ID: tipo entero, largo máximo 4 dígitos, $1 \leq ID \leq 9999$.
- NumTel: tipo entero, largo máximo 8 dígitos, $10000000 \leq NumTel \leq 99999999$.
- Region: tipo entero, $1 \leq Region \leq 15$

Los atributos están separados por un espacio y existe un registro por línea. El archivo debe ser indexado usando la representación binaria del atributo ID.

Salida

Indexación

Durante la indexación el programa debe generar un log (llamado **hashing.log**) donde se registran todas las operaciones generadas por el algoritmo de inserción. Considerando las posibles operaciones, el formato en el cual éstas se deben registrar, según corresponda, es el siguiente:

```
insert[clave]--> bucket[numBucket]           # la clave se inserta en bucket
insert[clave]--> bucket[numBucket]--> overflow # la clave se inserta en bucket de overflow
insert[clave]--> bucket[numBucket]--> split   # la clave se inserta en bucket luego de un split
```

Consultas

Luego de la indexación el programa deberá procesar un archivo (llamado **consultas.txt**) que contiene por cada fila un valor $1 \leq ID \leq 9999$. La salida de las consultas debe ser mostrada en pantalla en el siguiente formato (según corresponda):

```
clave[XXXX] --> no se encuentra en el índice
clave[XXXX] --> bucket[numBucket]
```

Parámetros del algoritmo

Factor de bloqueo = 4

Umbral = 75%

$d = 1$ (valor inicial)

$n = 2$ (valor inicial)

Restricciones e indicaciones

- La tarea debe ser codificada en Lenguaje C. No utilice funciones que no pertenecen al ANSI C. Utilice gcc para la compilación.
- Sólo se consideran las tareas que cumplan con las especificaciones planteadas.
- Sólo se recibirán tareas fuera de plazo dentro de las 24 horas siguientes a la fecha de entrega. Nota máxima es un 5.0
- La corrección de la tarea incluye una interrogación.

Fecha de entrega: Jueves 30 de Abril hasta las 24h00. Enviar código fuente tarea1-NombreApellido.c por email a ici343@inf.ucv.cl. El código debe contener como comentario el nombre del autor y su rut.