Estructuras de Datos y Algoritmos

Año 2022

Práctico 5: *Rebalse - Lista de Dos Niveles - Skip-List*

Ingeniería en Computación - Ingeniería en Informática

Ejercicio 1:

Se quiere almacenar la secuencia: $4371, 1323, 6173, 4199, 4344, 9679, 1989$ en una estructura de rebalse donde $M = 10$ y la función $h(x) = x$ **mod** 10.

a) Mostrar la correspondiente tabla de *rebalse separado* que se obtiene luego de insertar los elementos de la secuencia.

b) Mostrar la tabla de un *rebalse abierto lineal*, obtenida al insertar la misma secuencia.

c) Mostrar el *rebalse abierto cuadrático* que resulta de la inserción de los elementos de la secuencia de entrada.

d) Mostrar el rebalse abierto que se obtiene si el tratamiento es con *paso realeatorizado* y la función $p(x) = 7 - (x \textbf{ mod } 5)$.

e) Obtener, para las estructuras obtenidas en $a), b), c)$ y $d)$, los costos medios a posteriori de *localización exitosa*, tomando como función de costo *cantidad de baldes consultados*.

f) Obtener, para las estructuras obtenidas en $a), b), c)$ y $d)$, los costos medios a posteriori de *localización exitosa*, tomando como función de costo *cantidad de celdas consultadas*. ¿Existe alguna diferencia con lo calculado en el punto anterior?

En cada caso aclare las hipótesis que haya utilizado.

Ejercicio 2:

Suponga que tiene una tabla de distribución pseudo-aleatoria de siete baldes, y que la función h es $h(i) = i$ **mod** 7.

a) Mostrar el rebalse separado que se obtiene al insertar los elementos: 1, 8, 27, 125, 216 y 343.

b) Repita lo hecho en el punto **a)** pero utilizando un rebalse abierto lineal como almacenamiento.

c) Analizar si las estructuras obtenidas en **a)** y en **b)** cambiarían si se modifica el orden de la secuencia de entrada.

d) ¿Es posible mantener ordenadas las listas de la estructura del punto **a)**? Si su respuesta es afirmativa explique cuáles serían los beneficios de hacerlo, si no lo es explique por qué no es posible.

Ejercicio 3:

Se desea almacenar la siguiente secuencia de elementos: a, b, c, d, e, f, g, en una tabla de distribución pseudo-aleatoria siendo los valores de función para

$$h : X \longmapsto [0 \ldots 10]$$

$$h(a) = 3, \quad h(b) = 4, \quad h(c) = 10, \quad h(d) = 10, \quad h(e) = 3, \quad h(f) = 9, \quad h(g) = 3$$

a) Mostrar la tabla obtenida al insertar la secuencia dada, si el tratamiento del rebalse es:

i) Lineal
ii) Cuadrático
iii) Paso realeatorizado
iv) Realeatorizado total

Cuando se necesite la función de paso considere que:

$$p : X \longmapsto [1 \ldots 10]$$

$$p(a) = 1, \quad p(b) = 3, \quad p(c) = 6, \quad p(d) = 4, \quad p(e) = 3, \quad p(f) = 1, \quad p(g) = 2$$

y, cuando se necesite además el número de intento, considere que $h(x) = h(x, 1), \forall x \in X$ y para los demás intentos:

$$\begin{array}{llll} h(d,2) = 4, & h(d,3) = 6, & h(e,2) = 10, & h(e,3) = 9, \\ h(f,2) = 3, & h(f,3) = 6, & h(f,4) = 0, & h(g,2) = 8 \end{array}$$

b) Calcular el esfuerzo medio y máximo de localización *exitosa*, a posteriori, para cada uno de los casos del punto **a)**. Considere como función de costo *cantidad de baldes consultados*.

c) Calcular el esfuerzo medio y máximo de localización que *fracasa*, a posteriori, para los casos **i)** y **ii)** del punto **a)**. Considere como función de costo *cantidad de baldes consultados*

Ejercicio 4:

Desarrollar los operadores *LOCALIZACIÓN*, *EVOCACIÓN*, *ALTA* y *BAJA*, necesarios para administrar una distribución pseudo-aleatoria de datos si el tratamiento de rebalse es:

a) Lineal
b) Cuadrático
c) Realeatorizado total.
d) Paso realeatorizado
e) Separado

Ejercicio 5:

Analizar si es posible realizar un cálculo del esfuerzo medio de evocación exitosa, a posteriori, mientras se realizan las altas en una distribución pseudo-aleatoria de datos con tratamiento de rebalse lineal. Si considera que es posible, explique cómo lo haría.

Ejercicio 6:

Analizar y explicar una estrategia que permita ubicar los datos, al momento de un alta, de manera tal de minimizar la varianza de los esfuerzos de evocación exitosa. Suponga que está trabajando con una distribución pseudo-aleatoria de datos con tratamiento de rebalse lineal.

Ejercicio 7:

Proponga una posible estrategia para absorber, como *nunca usadas* o *vírgenes*, las celdas *libres* (utilizadas alguna vez), en una distribución pseudo-aleatoria de datos, con un tratamiento de rebalse lineal. Debe asegurarse de que la estructura siga funcionando correctamente.

Ejercicio 8:

Dada una relación $R \subseteq X \times Y$ almacenada en un direccionamiento directo (DD), sabiendo además que se cuenta con una función de enumeración e para X, desarrollar las rutinas que permitan operar sobre la estructura en cada uno de los siguientes casos:

a) Se ha realizado memorización previa y la relación es completa respecto de X.

b) No se ha realizado memorización previa y la relación es completa respecto de X.

c) Se ha realizado memorización previa y la relación no es completa respecto de X.

d) No se ha realizado memorización previa y la relación no es completa respecto de X.

Ejercicio 9:

Dada una relación $R \subseteq X \times Y$, donde X es asociante e Y la información asociada, almacenada en una Lista de dos Niveles, se pide analizar:

a) Sabiendo que para realizar la evocación de una nupla es necesaria una localización en el primer nivel de la estructura, seguida de una localización en el segundo nivel de la misma, ¿Qué método de búsqueda utilizaría para cada una de estas localizaciones?

b) Si las sublistas son de tamaño m, ¿por qué es necesario mantenerlas como mínimo con $\frac{m}{2}$ nuplas? ¿Hay alguna excepción?

Ejercicio 10:

Desarrollar en pseudo-código los operadores de *LOCALIZACIÓN*, *EVOCACIÓN*, *ALTA* y *BAJA*, para la administración de una Lista de dos Niveles. Aclarar las hipótesis que considere necesarias.

Ejercicio 11:

Analizar cuánto sería el esfuerzo máximo de alta a priori en una Lista de dos Niveles, medido en *cantidad de corrimientos*, si el nivel de descriptores tiene un extremo móvil y el de datos recircula sobre su espacio. Describir claramente el caso en que este máximo ocurre y plantear su costo asumiendo que m' es la cantidad de descriptores y m es el tamaño máximo de cada sublista, L el tamaño de los descriptores y l el tamaño de la nupla.

Ejercicio 12:

Si sabe que necesita almacenar N elementos en una Lista de dos Niveles, ¿Cómo organizaría los elementos en las distintas listitas del segundo nivel, si se desea tener la menor cantidad de descriptores posible?

a) ¿Qué beneficios provee una estructura organizada de ese modo?

b) ¿Qué problemas presenta?

c) Y si las organizara para tener la mayor cantidad de descriptores, ¿Cómo respondería a las preguntas anteriores?

Ejercicio 13:

Suponiendo que dispone de una Skip List inicialmente vacía, se pide:

a) Insertar la siguiente secuencia de elementos mostrando la estructura después de cada inserción:

55, 32, 132, 200, 861, 823, 937, 916, 524

Para ello suponga que:

- El máximo nivel posible es 3.

- En la rutina que genera el nivel de un elemento dada en teoría (ver Skip-Lis, pág.11), la evaluación de la condición "*Random( )*< $\frac{1}{2}$" produce la siguiente secuencia de Verdaderos-Falsos (F significa que el *Random* es mayor que $\frac{1}{2}$ y V que es menor):

$$F\ V\ F\ V\ V\ V\ F\ F\ V\ V\ F\ F\ F\ F\ V\ V\ V\ V\ V\ F\ \ldots$$

b) Insertar, en una Skip List inicialmente vacía, los mismos elementos del punto $a)$ pero en la siguiente secuencia:

32, 937, 55, 861, 200, 916, 524, 823, 132

asuma las mismas condiciones dadas en **a)**, mostrando la estructura final obtenida.

c) ¿Obtuvo la misma estructura? ¿Por qué?

d) ¿Los costos a posteriori de localizar con éxito cada elemento, considerando las celdas consultadas, se modifican? ¿Y los esfuerzos medios de localización exitosa?.

Ejercicio 14:

Para la Skip List obtenida en el ejercicio anterior, realizar la siguiente secuencia de bajas, mostrando como queda la estructura después de cada operación:

823, 32, 861, 132

Ejercicio 15:

Desarrollar los operadores de *LOCALIZACIÓN*, *EVOCACIÓN*, *ALTA* y *BAJA* que permitan administrar una Skip List, asumiendo la rutina que determina el nivel del elemento ya programada.