Lista de dos Niveles

Año 2024

Introducción

En esta asignatura tratamos de resolver el siguiente problema: dado un conjunto, cómo distribuimos los elementos del conjunto en una estructura de almacenamiento de forma tal de poder resolver eficientemente la localización de un elemento en la estructura. Planteamos como muy importante el resolver la localización, porque ella se usa en todas las rutinas que operarán sobre la estructura: *Pertenencia*, *Alta*, *Baja* y *Evocación Asociativa* (con asociante X y asociado Y, o sea, en la cual se aporta la componente X y se obtiene como respuesta la componente Y).

Hemos visto algunos tipos simples de estructuras de almacenamiento: listas desordenadas y ordenadas (secuenciales y vinculadas). De todas ellas la más eficiente hasta ahora fue una lista de alojamiento secuencial ordenada en la que podíamos aplicar un tipo de búsqueda llamada de *trisección* (o *bisección*). Ella permitía resolver la localización examinando sólo partes del conjunto y no la totalidad, utilizando la información brindada por el orden.

Mantener dicho conjunto en una lista secuencial ordenada nos permite tener un esfuerzo medio y máximo de localización (exitosa y no exitosa) de $O(\log_2 N)$ celdas consultadas, pero con ello tenemos un esfuerzo máximo de alta de $O(N)$ corrimientos de celdas. El esfuerzo de localización es muy bueno frente al que tenemos en las otras estructuras, pero el esfuerzo máximo de alta es malo. Por otra parte, en la lista desordenada tenemos un muy buen esfuerzo máximo de altas y bajas que es de $O(1)$, pero el esfuerzo máximo de localización (exitosa y no exitosa) no es bueno por ser $O(N)$. Por lo tanto, podemos tratar de encontrar una nueva estructura que guarde los beneficios de la localización de $O(\log_2 N)$ celdas consultadas de la lista secuencial ordenada, pero que logre mejorar su esfuerzo máximo de alta; es decir que el esfuerzo máximo de alta de la nueva estructura sea algo intermedio entre $O(1)$ (el de la lista desordenada) y $O(N)$ (el de la lista ordenada).

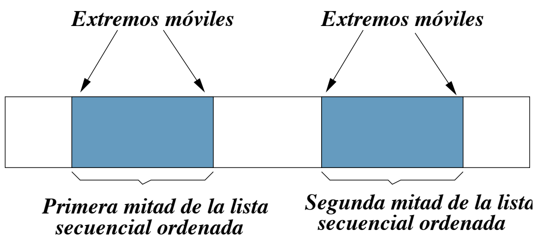
La primera mejora posible es mantener ambos extremos de la lista móviles (considerando el espacio disponible como circular), manteniendo así la localización de $O(\log_2 N)$ celdas consultadas y teniendo como máximo $\approx \dfrac{N}{2}$ corrimientos en cada modificación estructural.

En este caso, se baja la cantidad máxima de corrimientos por la posibilidad de elegir qué trozo de la lista se deberá correr para minimizar la cantidad de corrimientos. Es decir, se elige hacia qué extremo se debe realizar la menor cantidad necesaria de corrimientos de celdas para dejar la estructura correcta.

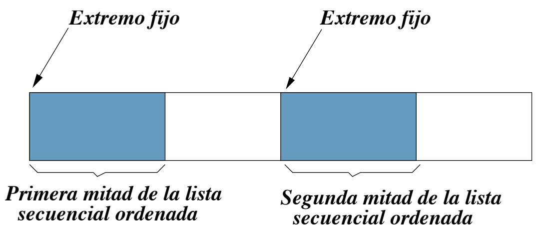
Otra manera para lograr nuestro objetivo es dividir nuestra lista en pedazos más chicos, distribuidos de manera tal que al realizar un alta podamos circunscribir en qué pedazo de la lista se debe realizar, y que la modificación necesaria para reflejar dicha alta en la estructura no deba realizar modificaciones a los otros pedazos de la lista. Además, como ya analizamos, mejoramos la cantidad de corrimientos necesarios para una modificación estructural si mantenemos ambos extremos de la lista móviles.

Supongamos, por ejemplo, que dividimos la lista secuencial ordenada original en dos trozos [1] , y mantenemos las dos sublistas (en el espacio disponible) de manera tal de permitirles crecimiento independiente a cada una. Como ya dijimos sus extremos podrían ser ambos móviles (**Caso 1**), para mejorar más la cantidad de corrimientos necesaria para una modificación estructural, o no (**Caso 2**).

Caso 1:



Caso 2:



En cualquier caso podemos seguir aplicando búsqueda binaria, porque dependiendo de la consigna elegida compararemos con el último elemento de la primera mitad o con el primero de la segunda mitad y en función de dicha comparación la búsqueda continuará, si debe hacerlo, en sólo una de las mitades igual que como funcionaba antes de repartirla. Por lo tanto, el esfuerzo máximo de localización de $O(\log_2 N)$ celdas consultadas se mantiene. Además, para realizar una modificación estructural en uno de los pedazos tendremos como máximo:

- $\approx \dfrac{N}{4}$ corrimientos de celdas si mantenemos ambos extremos de las listas móviles (**Caso 1**), o
- $\approx \dfrac{N}{2}$ si mantenemos sólo un extremo de las listas móvil (**Caso 2**).

O sea, bajamos al menos a la mitad el esfuerzo máximo de modificación estructural.

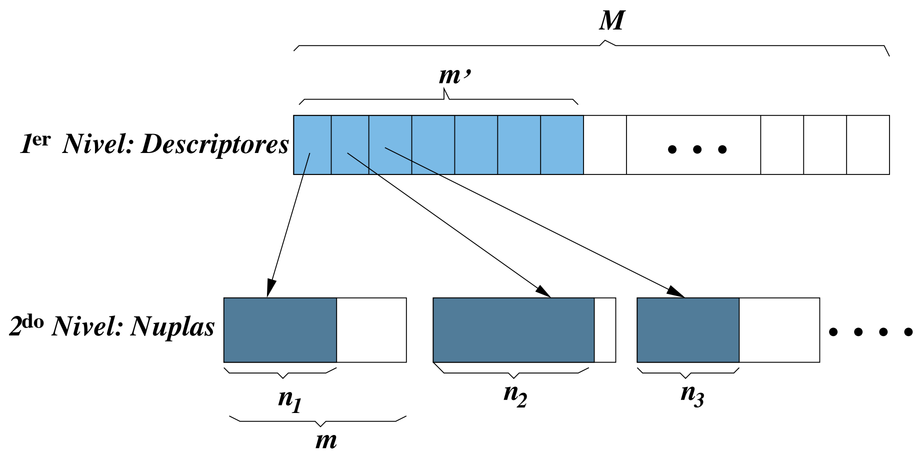
Queda claro que ahora necesitamos mantener información sobre dos trozos de lista y antes manteníamos sólo de una.

Se puede ver claramente que si continuamos dividiendo una lista en sublistas más pequeñas se mejorará el costo de las altas, y mientras en más pedazos se reparta se mejora más, porque bajará la cantidad de corrimientos.

[1] Recordar que para minimizar la varianza de los esfuerzos se deben mantener los trozos de tamaños muy similares. En este caso aproximadamente la mitad de los elementos en cada sublista.

Para mantener la principal ventaja de las listas secuenciales, cada una de las sublistas se debe mantener ordenada y además, si se quiere mantener baja la varianza del esfuerzo medio de localización, se deben mantener las listas lo más parejas posibles en cantidad de elementos presentes. Entonces, partimos la lista en trozos de igual tamaño y si podemos distinguir fácilmente cada sublista podremos pensar en aplicar búsqueda binaria sobre cada uno de los trozos.

Cada sublista debería ocupar a lo sumo un trozo. Entonces, cuando la lista se llena se la divide a la mitad, se deja una mitad en el trozo que ocupaba y se ubica la otra mitad en un nuevo trozo. Además, para evitar que esta operación provoque correr trozos, ellos se mantienen sin necesidad de secuencialidad (correr en un trozo de la lista no implica correr a los otros trozos). Para poder hacerlo así, se debe agregar otro nivel a la lista que mantenga la información de qué trozo sigue a cuál. Por lo tanto, la estructura descripta, llamada *Lista de dos Niveles*, sería como:



Si cada trozo de lista tiene capacidad para mantener a lo sumo m elementos; para mantener los trozos similares en tamaño colocamos la restricción de que cada trozo puede mantener como mínimo $\left\lfloor \dfrac{m+1}{2} \right\rfloor$ elementos.

Por lo tanto, cuando dividimos un trozo dejamos esa cantidad de elementos ocupando el trozo viejo y los restantes $\left\lceil \dfrac{m+1}{2} \right\rceil$ elementos los colocamos en el nuevo trozo.

En ese caso, para mantener la información correcta se debe actualizar el nivel que contiene la información sobre las sublistas.

Analicemos qué hacer en el alta. Para ello, por ejemplo, tenemos que considerar las siguientes cuestiones en el nivel de los elementos:

$$\begin{cases} \text{Hay lugar:} \qquad n_i \leq m-1 \\ \\ \text{No hay lugar} \end{cases}$$

$$\text{¿Cada lista recircula?} \begin{cases} \text{Sí} \\ \\ \text{No} \end{cases}$$

Si hay lugar en la lista se debe dar de alta en la posición que corresponda, dejándola libre mediante corrimientos de elementos. En caso de no haber lugar, se deberá crear una nueva lista y para ello se deben hacer corrimientos para trasladar los elementos a la otra lista y también para ubicar el nuevo elemento en el lugar correspondiente.

En cada posición de la lista de descriptores, que mantiene información sobre las sublistas, se debe mantener la descripción de la sublista correspondiente; o sea, el descriptor de esa sublista. Cada descriptor contiene todo lo necesario para describir completamente a una lista y también aquella información que pueda ayudar para las localizaciones. Si tenemos, por ejemplo, que las sublistas recirculan sobre el espacio y tienen ambos extremos móviles y la representación no varía en cada una de las sublistas, se mantendría:

a) Inicio: l_i

b) Fin: l_s

ya que necesito saber qué parte se está ocupando realmente, además de cuál es el trozo (dirección física de inicio del mismo).

Si la lista mantiene sólo un extremo móvil, o sea la lista es estable (anclada en un extremo), basta con mantener en el descriptor el límite superior junto con la dirección física de inicio del trozo correspondiente.

Además para evitar la referencia indirecta realizada en la búsqueda para ubicar la sublista podemos mantener algo que indique qué elementos se encuentran en la sublista correspondiente, por ejemplo, almacenar en el descriptor el primero o el último valor de X de todos aquellos elementos almacenados en dicha sublista.

Hemos planteado esta estructura para mejorar conjuntamente el esfuerzo de las altas, pero manteniendo la posibilidad de aplicar búsqueda binaria; por lo tanto, debemos calcular los esfuerzos para luego optimizarlos analizando qué parámetro del diseño los mejoraría.

Parámetros de la estructura:

M	:	espacio disponible para descriptores.
N	:	cantidad de elementos a alojar.
m	:	tamaño máximo para cada sublista de elementos.
m'	:	cantidad de descriptores en la estructura.
n_i	:	cantidad de elementos en la i-ésima sublista, $\left\lfloor \frac{m+1}{2} \right\rfloor \leq n_i \leq m$.
l	:	largo del elemento o nupla.
L	:	largo de un descriptor de lista.

en general, el tamaño de la nupla será mayor que el del descriptor; aunque puede haber situaciones particulares en que esto no sea así.

Análisis del peor caso para un alta

Supongamos mantener todas las listas (de primer y segundo nivel) con sólo un extremo móvil. Un análisis similar se puede hacer si otra es la manera de manejarlas.

Los costos serán medidos en cantidad de corrimientos.

El peor caso para un alta será entonces aquel que realice el mayor número de corrimientos posibles, tanto en el primero como en el segundo nivel. Por lo tanto, hay que encontrar aquel caso que, en conjunto, haga la mayor cantidad de corrimientos posibles en ambos niveles.

No se pueden analizar por separado los corrimientos máximos en cada nivel porque se podría llegar a un caso imposible.

Por ejemplo, el peor caso posible de corrimientos en el primer nivel (nivel de descriptores) sería que se deba dar de alta un descriptor que necesite ir antes del primer descriptor que se encuentra en el primer nivel, lo cual haría m' corrimientos de descriptores y ello significaría que se debió crear una sublista antes de la primera; pero de acuerdo a lo que ya hemos planteado, respecto de cómo se desdoblan las listas, esto no es posible.

Entonces, el peor caso posible para un alta de un elemento en una lista de dos niveles ocurre al dar de alta un elemento cuya componente x es menor que todas las ya presentes en la estructura, y a su vez en la primera sublista (que es en la que corresponde dar de alta un elemento teniendo ese x) no hay lugar disponible; o sea, la primera sublista tiene m elementos ya alojados. En este caso se deben realizar $m\ ' - 1$ corrimientos de descriptores y m corrimientos de elementos. En símbolos, y considerando el tamaño de cada uno de los objetos a correr obtenemos:

$$(m' - 1) \cdot L + m \cdot l \tag{1}$$

En este caso estamos en la siguiente situación:

1. Existen N elementos a instalar.
2. La primera lista contiene m elementos.
3. Las siguientes listas (salvo la última) contienen la menor cantidad posibles de elementos; o sea: $\left\lfloor \dfrac{m+1}{2} \right\rfloor$. Esto es para maximizar la cantidad de descriptores, lo que daría el peor caso de corrimientos en el primer nivel.
4. La última lista contiene al menos $\left\lfloor \dfrac{m+1}{2} \right\rfloor$, junto con todos los elementos restantes.

   Es decir, aquellos elementos que sobran porque no alcanzan para crear una lista más por otro desdoblamiento.

   Por lo tanto, la cantidad de elementos en la última lista sería: $\left\lfloor \dfrac{m+1}{2} \right\rfloor \leq n_{m'} \leq m$.

Por lo tanto, es verdadera la siguiente desigualdad:

$$m + (m' - 1) \cdot \left\lfloor \frac{m+1}{2} \right\rfloor \leq N$$

porque hay una lista con m elementos y considerando el resto de las listas con la mínima cantidad de elementos para maximizar la cantidad de descriptores $\left(\left\lfloor\frac{m+1}{2}\right\rfloor\right)$ la última podría tener más elementos, pero en total pueden haber a lo sumo N elementos.

De aquí se puede obtener, despejando m' que:

$$m' \leq \frac{N-m}{\left\lfloor\frac{m+1}{2}\right\rfloor} + 1$$

pero, queremos el mayor entero m' tal que haga a esta ecuación verdadera; por lo tanto, debemos aplicar la función piso a la parte derecha de la desigualdad para quedarnos con un entero que la haga verdadera:

$$m' \leq \left\lfloor \frac{N-m}{\left\lfloor\frac{m+1}{2}\right\rfloor} + 1 \right\rfloor$$

y luego quedarnos con igualdad, porque así obtenemos el mayor valor de todos aquellos que la hacen verdadera. Entonces sería:

$$m' = \left\lfloor \frac{N-m}{\left\lfloor\frac{m+1}{2}\right\rfloor} + 1 \right\rfloor$$

Para simplificar esta fórmula podemos reemplazar $\left\lfloor\frac{m+1}{2}\right\rfloor$ por su equivalente $\left\lceil\frac{m}{2}\right\rceil$ y obtenemos:

$$m' = \left\lfloor \frac{N-m}{\left\lceil\frac{m}{2}\right\rceil} + 1 \right\rfloor \tag{2}$$

Ahora reemplazamos el valor de m' en la fórmula (1) del esfuerzo máximo de alta obteniendo así:

$$\left(\left(\left\lfloor \frac{N-m}{\left\lceil\frac{m}{2}\right\rceil} + 1 \right\rfloor\right) - 1\right) \cdot L + m \cdot l \tag{3}$$

Queremos minimizar este esfuerzo respecto de m, ya que m es el único de los parámetros que no viene ya impuesto [2]. Como esta función es discontinua (por la aplicación de la función piso y techo), un recurso práctico para poder encontrar el mínimo de esta función respecto de m es reemplazarla por una función continua.

[2] Los otros valores que intervienen aquí o vienen impuestos por el problema, como l, o no permiten mucho margen de cambio, como L.

Para convertir a la función en continua hacemos uso de:

$$m' \approx \frac{N-m}{\frac{m}{2}} + 1 \tag{4}$$

y al reemplazar en la fórmula (3) del esfuerzo máximo nos queda:

$$\left(\left(\frac{N-m}{\frac{m}{2}} + 1 \right) - 1 \right) \cdot L + m \cdot l \tag{5}$$

Simplificando (5) quedaría ahora:

$$2 \cdot L \cdot \left(\frac{N-m}{m} \right) + m \cdot l \tag{6}$$

Por lo tanto, lo que queremos obtener es:

$$\min_{m} \left(2 \cdot L \cdot \left(\frac{N-m}{m} \right) + m \cdot l \right)$$

Para obtener un mínimo para esta función primero tomamos la primera derivada respecto de m e igualamos a 0 (recordando que en un extremo de una función la primera derivada se anula):

$$2 \cdot L \cdot \left(\frac{-1}{m} \right) - 2 \cdot L \cdot \left(\frac{N-m}{m^2} \right) + l = 0$$

Ahora podemos sacar común denominador m^2 y quedaría:

$$\frac{-2 \cdot L \cdot m - 2 \cdot L \cdot (N-m) + l \cdot m^2}{m^2} = 0$$

distribuyendo, considerando que para que esta fórmula valga 0 su numerador debe ser 0 y simplificando obtenemos:

$$\begin{aligned} -2 \cdot L \cdot m - 2 \cdot L \cdot (N-m) + l \cdot m^2 &= 0 \\ -2 \cdot L \cdot N + l \cdot m^2 &= 0 \end{aligned}$$

despejando ahora m llegamos a que:

$$\begin{aligned} m^2 &= \left(\frac{2 \cdot L}{l} \right) \cdot N \\ m &= \sqrt{\left(\frac{2 \cdot L}{l} \right) \cdot N} \end{aligned} \tag{7}$$

Para verificar que en ese punto la función tiene un mínimo y no un máximo, debemos comprobar si la segunda derivada de $2 \cdot L \cdot \left(\frac{N-m}{m} \right) + m \cdot l$, para $m = \sqrt{\left(\frac{2 \cdot L}{l} \right) \cdot N}$, es positiva:

$$\left(2 \cdot L \cdot \left(\frac{N-m}{m} \right) + m \cdot l \right)'' = \left(-\frac{2 \cdot L \cdot N}{m^2} + l \right)' = \frac{4 \cdot L \cdot N \cdot m}{m^4}$$

si reemplazamos en la segunda derivada el valor de $m = \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}$ y considerando que $N > 0$ se tiene que:

$$\frac{4 \cdot L \cdot N \cdot m}{m^4} = \frac{4 \cdot L \cdot N}{m^3} = \frac{\overbrace{4 \cdot L \cdot N}^{>0}}{\underbrace{\left(\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}\right)^3}_{>0}}$$

$$\Rightarrow \frac{4 \cdot L \cdot N}{\left(\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}\right)^3} \geq 0$$

con lo cual queda demostrado que en $m = \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}$ la función $2 \cdot L \cdot \left(\frac{N - m}{m}\right) + m \cdot l$ tiene un mínimo.

Ahora podemos reemplazar el valor de m en la fórmula (6) del esfuerzo, obteniendo:

$$\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} \cdot l + 2 \cdot \left[\frac{N - \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}}{\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}}\right] \cdot L$$

Introduciendo a l dentro de la raíz cuadrada y distribuyendo obtenemos:

$$\sqrt{2 \cdot L \cdot l \cdot N} + \frac{2 \cdot L \cdot N - 2 \cdot L \cdot \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}}{\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}}$$

Eliminamos la raíz cuadrada del denominador, y distribuyendo y simplificando:

$$\sqrt{2 \cdot L \cdot l \cdot N} + \frac{2 \cdot L \cdot N - 2 \cdot L \cdot \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}}{\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}} \cdot \frac{\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}}{\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}} =$$

$$\sqrt{2 \cdot L \cdot l \cdot N} + \frac{2 \cdot L \cdot N \cdot \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} - 2 \cdot L \cdot \left(\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}\right)^2}{\left(\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}\right)^2} =$$

$$\sqrt{2 \cdot L \cdot l \cdot N} + \frac{2 \cdot L \cdot N \cdot \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}}{\frac{2 \cdot L \cdot N}{l}} - 2 \cdot L =$$

$$\sqrt{2 \cdot L \cdot l \cdot N} + l \cdot \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} - 2 \cdot L =$$

$$\sqrt{2 \cdot L \cdot l \cdot N} + \sqrt{\left(\frac{2 \cdot L \cdot l^2}{l}\right) \cdot N} - 2 \cdot L =$$

$$\sqrt{2 \cdot L \cdot l \cdot N} + \sqrt{2 \cdot L \cdot l \cdot N} - 2 \cdot L =$$

$$2 \cdot \sqrt{2 \cdot L \cdot l \cdot N} - 2 \cdot L =$$

$$2 \cdot \sqrt{2 \cdot L \cdot l} \cdot \sqrt{N} - 2 \cdot L \tag{8}$$

En la primera igualdad distribuimos la raíz cuadrada respecto de la diferencia, en la segunda simplificamos las raíces cuadradas, en la tercera simplificamos el segundo término, en la cuarta introducimos a l dentro de la raíz cuadrada para eliminar el denominador, en la quinta simplificamos el segundo término, en la sexta asociamos los términos iguales y en la última separamos en el primer término lo que es constante de lo que es una función de N (el número de elementos del conjunto). Así, llegamos finalmente a la fórmula del esfuerzo máximo (8) medido en cantidad de corrimientos.

Por lo tanto, el esfuerzo máximo de alta es $O(\sqrt{N})$ corrimientos. Podemos recordar que $\log_2 N \in O(\sqrt{N})$ (porque $\sqrt{N}$ crece más rápidamente que $\log_2 N$), pero es mucho mejor que $O(N)$.

Análisis del peor caso para una baja

Supongamos, como antes, que se representan todas las listas (de primer y segundo nivel) con sólo un extremo móvil y que la función de costo utilizada es la cantidad de corrimientos [3]. Un análisis similar se puede hacer si otra es la manera de manejarlas las listas (por ejemplo: con los dos extremos móviles y recirculación).

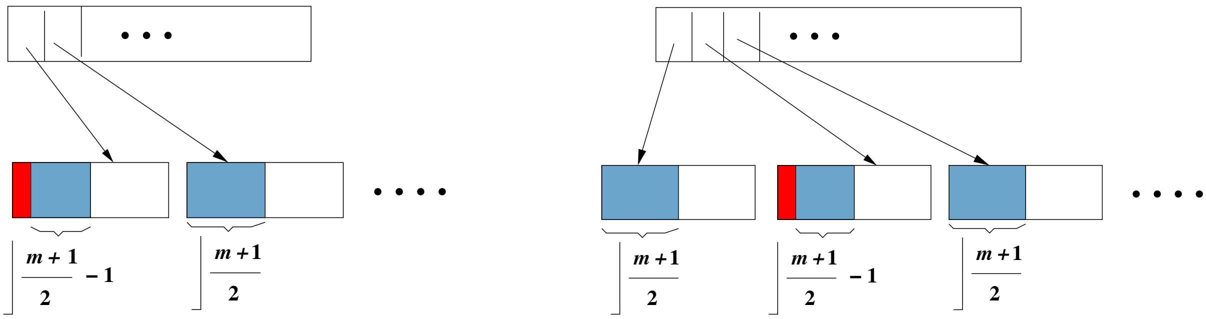
El peor caso para un baja sería entonces aquel que realice el mayor número de corrimientos posibles, tanto en el primero como en el segundo nivel. Por lo tanto, hay que determinar el caso que, en conjunto, haga la mayor cantidad de corrimientos posibles en ambos niveles. Como en el caso de las altas, no se pueden analizar por separado los corrimientos máximos en cada nivel porque se podría llegar a un caso imposible.

Para que se deban hacer corrimientos de descriptores la sublista en la que se realiza la baja debe quedar con $\left\lfloor \frac{m+1}{2} \right\rfloor - 1$ elementos, y las sublistas vecinas deben tener exactamente $\left\lfloor \frac{m+1}{2} \right\rfloor$ para que no puedan pasarle un elemento de manera de cubrir, en la sublista donde se produjo la baja, el mínimo número de elementos permitido sin perder ellas dicho mínimo. De esta manera la sublista que no llega al mínimo de elementos deberá ser eliminada, por lo que también se elininará su descriptor. Además, las restantes sublistas deberían contener $\left\lfloor \frac{m+1}{2} \right\rfloor$ elementos cada una, para maximizar el número de

[3] Para el análisis del peor caso de una baja se consideró como base un borrador del desarrollo realizado por la alumna Marina Álvarez.

descriptores. Así mismo, para correr la mayor cantidad de descriptores, la sublista donde se realice la baja y cuyo descriptor debería eliminarse, tendría que estar al comienzo de la lista de descriptores.

Las siguientes figuras ilustran los casos que podrían representar los peores casos de baja. Aquí se muestra la eliminación del primer elemento de la primera sublista (izquierda) y del primero de la segunda (derecha) respectivamente. Como puede observarse, en ambos casos la cantidad de nuplas que deben correrse es la misma, aunque en el caso en que se elimina el menor elemento de la primera sublista se debería correr una mayor cantidad de descriptores en el primer nivel que si se eliminara el menor elemento de la segunda.



Entonces, el peor caso de baja sucedería al eliminar el menor elemento del conjunto; es decir el primer elemento de la primer sublista. Luego de la eliminación ésta quedará con $\left\lfloor \frac{m+1}{2} \right\rfloor - 1$ elementos por lo que debería pedir a su vecina una nupla para recuperar la cantidad mínima de elementos permitida. Dado que no es posible pasar ningún elemento a la primera sublista desde la sublista adyacente, ésta se deberá eliminar por falta de elementos obligando a eliminar también a su descriptor.

Al eliminar el primer elemento de la primera sublista, y como tiene que desaparecer una sublista, se podría proceder de dos maneras:

1. Pasar los $\left\lfloor \frac{m+1}{2} \right\rfloor - 1$ elementos de la primera sublista a la segunda: lo que implica correr los $\left\lfloor \frac{m+1}{2} \right\rfloor$ elementos que tiene la segunda sublista hacia la derecha $\left\lfloor \frac{m+1}{2} \right\rfloor - 1$ posiciones, es decir el último elemento se moverá a la posición $k = \left\lfloor \frac{m+1}{2} \right\rfloor - 1 + \left\lfloor \frac{m+1}{2} \right\rfloor$, el penúltimo a la posición $k-1$, y así sucesivamente para hacer lugar a los $\left\lfloor \frac{m+1}{2} \right\rfloor - 1$ elementos de la primera sublista, y luego pasar de manera ordenada estas $\left\lfloor \frac{m+1}{2} \right\rfloor - 1$ nuplas [4]. A continuación, para eliminar el descriptor de la primera sublista en el primer nivel, se deberían correr $m' - 1$ descriptores una posición hacia la izquierda.
2. Pasar los $\left\lfloor \frac{m+1}{2} \right\rfloor$ elementos de la segunda sublista a la primera: lo que implica primero correr cada uno de los $\left\lfloor \frac{m+1}{2} \right\rfloor - 1$ elementos de la lista en la que se dio de baja una posición hacia la izquierda, y luego pasar los $\left\lfloor \frac{m+1}{2} \right\rfloor$ elementos de la segunda a continuación de los que están

[4] El valor de k será $m-1$ si m es par y $k = m$ si m es impar.

en la primer sublista. Posteriormente, para eliminar el descriptor de la segunda sublista, se deben correr los $m' - 2$ descriptores del primer nivel una posición hacia la izquierda.

Claramente ambas opciones deben correr la misma cantidad de nuplas en el segundo nivel, pero la segunda opción correría un descriptor menos, por lo tanto se debería proceder como se detalla en 2. Así se obtendría el siguiente esfuerzo máximo de baja:

$$(m' - 2) \cdot L + \left(\left\lfloor \frac{m+1}{2} \right\rfloor + \left\lfloor \frac{m+1}{2} \right\rfloor - 1 \right) \cdot l \tag{9}$$

Habiendo ya obtenido el valor de $m = \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}$, que minimiza el esfuerzo máximo de alta, se debe determinar el valor de m' para el peor caso de baja.

$$\begin{aligned} \left\lfloor \frac{m+1}{2} \right\rfloor m' &\leq N \\ m' &\leq \frac{N}{\left\lfloor \frac{m+1}{2} \right\rfloor} \end{aligned}$$

pero, queremos el mayor entero m' tal que haga a esta ecuación verdadera. Llamemos $m^* = \frac{N}{\left\lfloor \frac{m+1}{2} \right\rfloor}$, claramente m^* puede ser un número real. Sin embargo, el valor de m' debe ser un número entero z que cumpla con ser $z \leq m^*$ y que ese número sea lo más grande posible que cumpla la condición, para lo cual debemos aplicar la función piso: $z = \lfloor m^* \rfloor \leq m^*$. Pero, como queremos que m' sea el valor entero más grande que satisfaga la condición $m' \leq z = \lfloor m^* \rfloor \leq m^*$, tendríamos que $m' = z = \lfloor m^* \rfloor \leq m^*$ y así $m' = \left\lfloor \frac{N}{\left\lfloor \frac{m+1}{2} \right\rfloor} \right\rfloor$.

Para simplificar, tal como lo hicimos en el análisis del peor caso de alta, se puede utilizar que $\left\lfloor \frac{m+1}{2} \right\rfloor$ es equivalente a $\left\lceil \frac{m}{2} \right\rceil$.

$$\begin{aligned} m' &\leq \left\lfloor \frac{N}{\left\lfloor \frac{m+1}{2} \right\rfloor} \right\rfloor \\ m' &\leq \left\lfloor \frac{N}{\left\lceil \frac{m}{2} \right\rceil} \right\rfloor \end{aligned}$$

Ahora reemplazamos el valor de m' en la fórmula 9 del esfuerzo máximo de baja obteniendo así:

$$\left(\left\lfloor \frac{N}{\left\lceil \frac{m}{2} \right\rceil} \right\rfloor - 2 \right) \cdot L + \left(\left\lceil \frac{m}{2} \right\rceil + \left\lceil \frac{m}{2} \right\rceil - 1 \right) \cdot l$$

Podemos observar que si m es impar $\left(\left\lceil\frac{m}{2}\right\rceil + \left\lceil\frac{m}{2}\right\rceil - 1\right) = m$ y en caso contrario si m es par $\left(\left\lceil\frac{m}{2}\right\rceil + \left\lceil\frac{m}{2}\right\rceil - 1\right) = m - 1$.

Por lo tanto, para simplificar dejaremos de lado la aplicción de las funciones de piso y techo y como estamos analizando el peor caso, consideraremos que $\left(\left\lceil\frac{m}{2}\right\rceil + \left\lceil\frac{m}{2}\right\rceil - 1\right) = m$.

$$\left(\left\lfloor\frac{N}{\left\lceil\frac{m}{2}\right\rceil}\right\rfloor - 2\right) \cdot L + \left(\left\lceil\frac{m}{2}\right\rceil + \left\lceil\frac{m}{2}\right\rceil - 1\right) \cdot l \approx \left(\frac{N}{\frac{m}{2}} - 2\right) \cdot L + m \cdot l = \left(\frac{2 \cdot N}{m} - 2\right) \cdot L + m \cdot l$$

y sacando común denominador y distribuyendo el producto:

$$\left(\frac{2 \cdot N}{m} - 2\right) \cdot L + m \cdot l = \frac{2 \cdot N \cdot L - 2 \cdot m \cdot L}{m} + m \cdot l$$

reemplazando el valor de $m = \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}$ se obtiene:

$$\frac{2 \cdot N \cdot L - 2 \cdot m \cdot L}{m} + m \cdot l = \frac{2 \cdot N \cdot L - 2\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} \cdot L}{\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}} + \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} \cdot l$$

para eliminar la raíz del denominador, racionalizamos multiplicando y dividiendo por $\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}$:

$$\frac{2 \cdot N \cdot L - 2\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} \cdot L}{\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}} \cdot \frac{\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}}{\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}} + \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} \cdot l =$$

$$\frac{2 \cdot N \cdot L \cdot \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} - 2 \cdot L \left(\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}\right)^2}{\left(\sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N}\right)^2} + \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} \cdot l =$$

$$\frac{2 \cdot N \cdot L \cdot \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} - 2 \cdot L \cdot \left(\frac{2 \cdot L}{l}\right) \cdot N}{\left(\frac{2 \cdot L}{l}\right) \cdot N} + \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} \cdot l =$$

$$\frac{2 \cdot N \cdot L \cdot l \cdot \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} - 2 \cdot L \cdot l \cdot \left(\frac{2 \cdot L}{l}\right) \cdot N}{2 \cdot L \cdot N} + \sqrt{\left(\frac{2 \cdot L}{l}\right) \cdot N} \cdot l =$$

introduciendo los términos l en las raíces y simplificando y reordenando términos se obtiene:

$$\frac{2 \cdot L \cdot N \cdot \sqrt{2 \cdot L \cdot l \cdot N} - 2 \cdot L \cdot 2 \cdot L \cdot N}{2 \cdot L \cdot N} + \sqrt{2 \cdot L \cdot l \cdot N} =$$

$$\sqrt{2\cdot L\cdot l\cdot N} - 2\cdot L + \sqrt{2\cdot L\cdot l\cdot N} =$$
$$2\cdot\sqrt{2\cdot L\cdot l\cdot N} - 2\cdot L = \left(2\cdot\sqrt{2\cdot L\cdot l}\right)\cdot\sqrt{N} - 2\cdot L \in O(\sqrt{N})$$

Por lo tanto, el esfuerzo máximo de baja es $O(\sqrt{N})$ corrimientos, lo que es mucho mejor que $O(N)$ corrimientos en la lista secuencial ordenada.

Puede observarse que el esfuerzo máximo de baja coincide con el esfuerzo máximo de alta previamente obtenido: $\left(2\cdot\sqrt{2\cdot L\cdot l}\right)\cdot\sqrt{N} - 2\cdot L$, a pesar de que estamos considerando que implica mover un descriptor menos:

Esf. Máximo de Alta	**Esf. Máximo de Baja**
$(m'-1)\cdot L + m\cdot l$	$(m'-2)\cdot L + m\cdot l$

pero se debe considerar que en el peor caso de baja consideramos tener un descriptor más que en el peor caso de alta porque:

- Baja: todas las listas lo más vacías posibles.
- Alta: primera lista llena y el resto lo más vacías posibles.

Análisis del peor caso de búsqueda

Analicemos ahora el esfuerzo máximo de búsqueda exitosa. En el primer nivel (nivel de descriptores) sólo podemos aplicar búsqueda por bisección ya que no podemos preguntar a ese nivel si un elemento, con un determinado valor de X, se encuentra porque en ese nivel no hay elementos sino descriptores de listas; pero sí podemos identificar en cuál de las sublistas podría estar el elemento buscado. En el segundo nivel (nivel de elementos) podemos aplicar la búsqueda binaria en cualquiera de sus dos modalidades.

Veamos por ejemplo cuál sería el esfuerzo máximo de localización exitosa si en el segundo nivel aplicáramos trisección:

Esf. máx. (Bisección en el 1^{er} Nivel) + *Esf. máx.* (Trisección en el 2^{do} Nivel)

Recordando que el esfuerzo máximo de localización exitosa, teniendo N elementos en la lista y considerando como función de costo cantidad de celdas consultadas, en cada caso era:

$$\textit{Esf. máx.}_{\text{ Trisección}}(N):\quad \lfloor \log_2 N \rfloor + 1$$

$$\textit{Esf. máx.}_{\text{ Bisección}}(N):\quad \lceil \log_2 N \rceil + 1$$

Por lo tanto, sustituyendo el valor de m' obtenido en (4), obtendríamos en este caso:

$$\left(\left\lceil \log_2\left(\frac{N-\sqrt{\left(\frac{2\cdot L}{l}\right)\cdot N}}{\frac{\sqrt{\left(\frac{2\cdot L}{l}\right)\cdot N}}{2}}\right)\right\rceil + 1\right) + \left(\lfloor \log_2 m \rfloor + 1\right)$$

pero debemos tener en cuenta además, que en realidad en la búsqueda en el primer nivel no se pretende encontrar el elemento a la salida de dicha búsqueda, sino sólo identificar una sublista, así habrá una pregunta por igual que no necesitamos hacer. Por lo tanto, en la fórmula desaparecería un “+1”.

Podemos razonarlo también de otra manera que hace más sencilla la visualización de dicho máximo. Para ello, hacemos las siguientes aproximaciones: $\lfloor \log_2 m \rfloor + 1 \approx \lceil \log_2 m \rceil$ y además $m' \cdot m \approx 2 \cdot N$; y así, llegamos a la siguiente expresión:

$$\lceil \log_2 m' \rceil + \lfloor \log_2 m \rfloor + 1 \quad \approx \quad \lceil \log_2 (m' \cdot m) \rceil \quad \approx \quad \lceil \log_2 (2 \cdot N) \rceil \quad \approx \quad \lceil \log_2 N \rceil + 1$$

Claramente se ve que dicho esfuerzo se mantiene en $O(\log N)$. Analizando más en detalle, y teniendo en cuenta las aproximaciones realizadas, podemos observar que con a lo sumo una consulta a celda más que lo que obtendríamos si hubiéramos aplicado trisección a la lista completa (con los N elementos) conseguimos buscar exitosamente un elemento en la lista de dos niveles.

Si aplicásemos bisección en ambos niveles obtendríamos el siguiente esfuerzo máximo de búsqueda:

$$\lceil \log_2 m' \rceil + \lceil \log_2 m \rceil + 1 \quad \approx \quad \lceil \log_2 (m' \cdot m) \rceil + 1 \quad \approx \quad \lceil \log_2 (2 \cdot N) \rceil + 1 \quad \approx \quad \lceil \log_2 N \rceil + 2$$

Aquí, se resuelve la búsqueda exitosa en la lista de dos niveles aproximadamente consultando a lo sumo una celda más que las que hubiéramos consultado al realizar bisección sobre una lista secuencial ordenada con los N elementos.

Reconocimientos

El presente apunte se realizó tomando como base notas de las clases del **Ing. Hugo Ryckeboer** en la Universidad Nacional de San Luis.