Análisis y Diseño de Algoritmos
Tablas de Hash

Guillermo Morales-Luna

Arturo Díaz Pérez

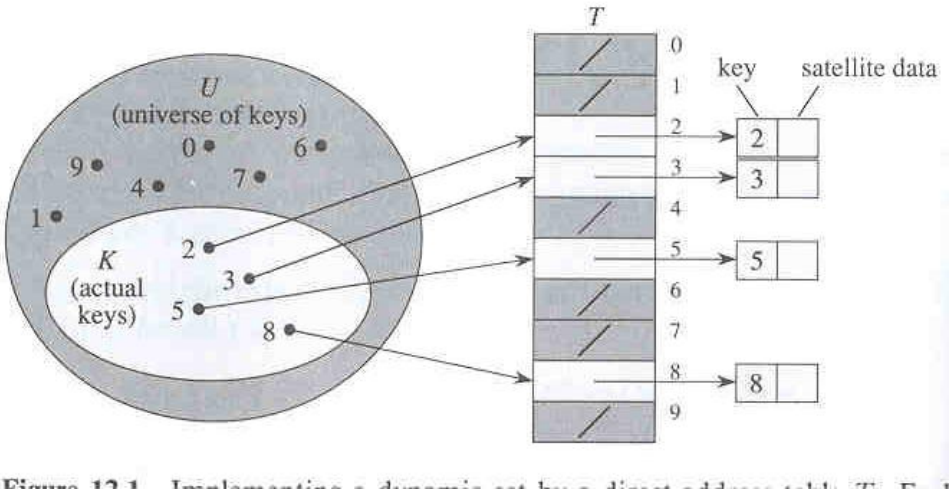
CONTENIDO

Dispersión

- Un *diccionario* es una estructura de datos abstracta para conjuntos dinámicos en los cuales se aplican las operaciones INSERCION, SUPRESION y BUSQUEDA.
- El uso de arreglos para almacenar estructuras de datos dinámicas es simple, sin embargo,
  - Los arreglos son de tamaño fijo
  - Normalmente se desperdicia espacio de memoria
- *Dispersión (Hash)*
  - Ubica registros en función de índices, o llaves, eficientemente.

Direccionamiento Directo

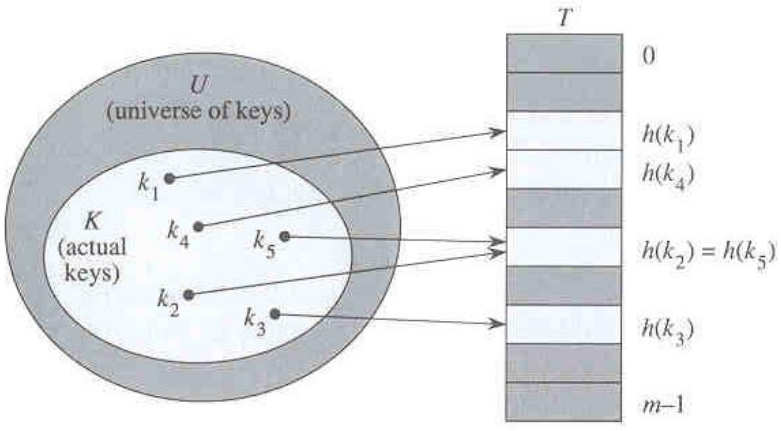
- Supongamos que se tiene un universo de llaves U razonablemente pequeño.
  - $U = 0, 1, \ldots, n-1$, donde n no es muy grande
  - No existen llaves duplicadas
- En el arreglo $T[1 \ldots n-1]$ cada posición se utiliza para representar una llave en el universo. Si el diccionario, $S \subset U$
  - si $i \notin S$, entonces, $T[i] = nil$
  - si $i \in S$, entonces, $T[i]$ proporciona un apuntador al registro.
- Espacio requerido: $\Theta(\text{card}(T))$
- Inserciones, supresiones y búsquedas toman tiempo $O(1)$.



Implementación de un conjunto dinámico mediante una tabla de dispersión T.

Tablas de Dispersión

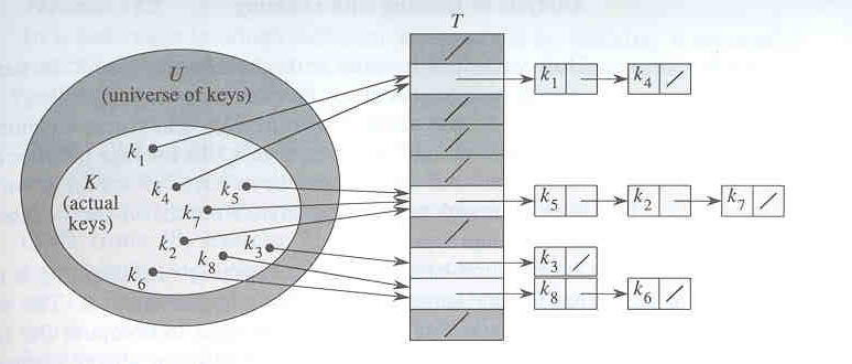
- La cardinalidad de U debe ser suficientemente pequeña para que el direccionamiento directo sea práctico
- Sean U: conjunto de llaves
- $T = [\![0, m-1]\!]$: direcciona elementos de U, $\mathbf{m} << \mathbf{n}$.
- $h : U \to T$: función *de dispersión*
  - T: *tabla de dispersión.*
- $m < n \Rightarrow$ no hay una función inyectiva $U \to T$,
- Dos llaves $u_1, u_2 \in U$ *colisionan* bajo h si $h(u_1) = h(u_2)$.



La función de dispersión h mapea las llaves a las ranuras de la tabla de dispersión.

Manejo de colisiones *por encadenamiento*

T: estructura de arreglo de listas ligadas; $\forall i \in T$, $T(i)$ lista ligada formada por llaves asociadas a i.



Análisis de las Tablas de Dispersión con Encadenamiento

- La *razón de carga* es $\alpha(n, m) = \frac{m}{n}$.
- En el peor caso todas las llaves caen en una ranura así que inserciones, supresiones y búsquedas toman un tiempo $\Theta(n)$ más el tiempo que tarde en calcular la función de dispersión.
  - Supongamos que calcular $h(i)$ toma un tiempo $O(1)$
- El comportamiento promedio de las tablas de dispersión depende de la efectividad de h para distribuir el conjunto de llaves.
- Se tiene una *dispersión uniforme* cuando $\text{card}(h^{-1}(i))$ es lo más cercano a la *razón de carga*.

Observación 1 *α: razón de carga de dispersión uniforme h, y manejo de colisiones por encadenamiento. Entonces,*

1. *Búsqueda no exitosa de llave tarda, en promedio, $\Theta(1+\alpha)$.*
2. *Búsqueda exitosa de llave tarda, en promedio, $\Theta(1+\alpha)$.*

En efecto, sea $u \in U$. En tiempo constante se calcula $i = h(u)$.

1. Dispersión uniforme, luego α el'tos en $I(i)$. Si búsqueda no exitosa, necesariamente se revisaron todos el'tos en $I(i)$. Tiempo $\Theta(1+\alpha)$.
2. ξ: variable aleatoria con valor una llave elegida uniformemente. $\forall u \in U:\ \text{prob}(\xi = u) = \frac{1}{n}$. Si u j-ésima llave e $i = h(j)$, antes de j cayeron, en $I(i)$, $(j-1)/m$ llaves. Por tanto, esperanza de "tiempo en encontrar":

$$\sum_{j=1}^{n} \text{prob}(\xi = u_j)\left(1 + \frac{j-1}{m}\right) = 1 + \frac{1}{nm}\sum_{j=1}^{n}(j-1) = 1 + \frac{\alpha}{2} - \frac{1}{2m} = \Theta(1+\alpha).$$

Funciones de dispersión

Si m es la dimensión de una tabla de dispersión, entonces,
$0 \le h(u_j) < m$

Se busca que $h : U \to I$ con
$\forall i \in I:\ \sum \text{prob}\{j \in U | h(j) = i\} \approx 1/m.$

Por ejemplo, si $(r_j)_{j=1}^{n}$: números aleatorios uniformes en $[0,1[$,
$h : u_j \mapsto \lfloor r_j \cdot m \rfloor$.

Consideraremos en esta presentación que $U \subset \mathbb{N}$.

Ejemplo. Sean u_j valores enteros de 10 dígitos decimales.
$h : u_j \mapsto k$ dígitos centrales de $u_j \times u_j$

Si $k = 3$, entonces, $m = 1000$.

Método de la división

$$h : k \mapsto k \bmod m \tag{1}$$

Algunos valores de m son mejores que otros.

- Si $m = 2^k$, el efecto de h es quedarse con los k *bits* menos significativos. Para evitar esto, utilizar módulos $m = 2^k - 1$, o más generalmente $m = b^k - 1$, b: base de representación.
- En general, se quiere evitar valores de m que dividan a $b^k \pm a$, donde k y a son números pequeños, b la base de representación.

- Sugerencia es elegir $m = p$, p: primo cercano a un factor α.
- La elección de m como primo cercano a α evita sesgos de representación.
- Para otro módulo $m > 1$ se puede sesgar un aumento de colisiones.

Método de la multiplicación

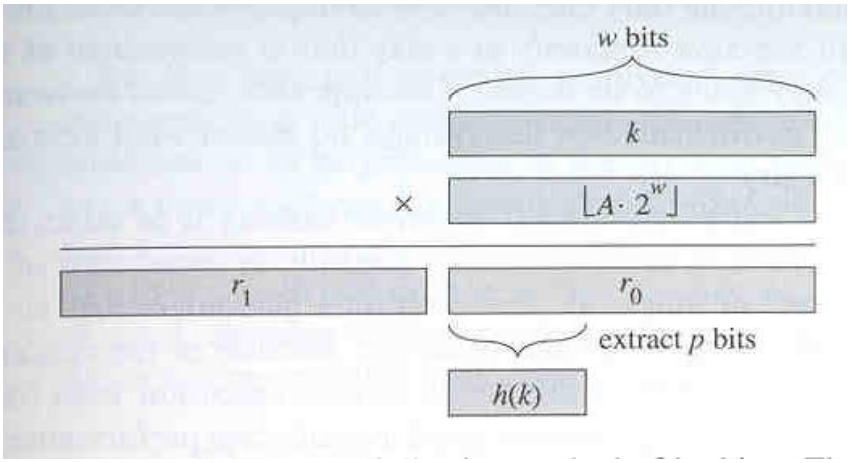
Sea $\cdot \bmod 1 : x \mapsto x - \lfloor x \rfloor$: *parte fraccionaria* de enteros. Sea w el tamaño de la palabra de una computador. Ej. 2^32, el enter A es la fracción A/w. El método es elegir una constante enter A prima relativa a w, y

$$h : k \mapsto \lfloor m((\frac{A}{w}k) \bmod 1) \rfloor \tag{2}$$

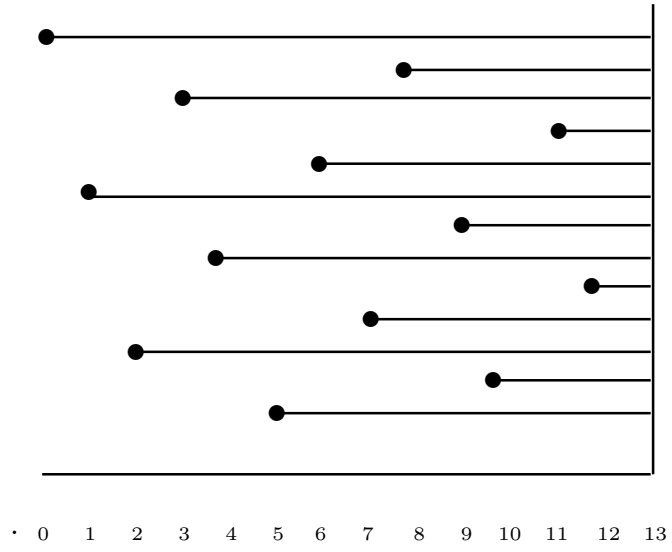
Si m es una potencia de 2, se elige A tal que $h(k)$ consiste de los bits más significativos de la mitad menos significativa del producto Ak.

El método multiplicativo convierte una progresión aritmética, $k, k + d, k + 2d, \ldots$, en una progresión aritmética aproximada, $h(k), h(k + d), h(k + 2d), \ldots$ de valores de hash distintos.

Método de la multiplicación



Sea A/w la razón áurea $\phi^{-1} = \frac{1}{2}(\sqrt{5}-1)$. $h(0), h(1), h(2), \ldots$, se mapean a $\{0\}, \{\phi^{-1}\}, \{2\phi^{-1}\}, \ldots$



Dispersión universal

Elegir una función de dispersión particular para cada llave.

$\mathcal{H} \subset [\![0, m-1]\!]^U$: colección (finita) de funciones.

$\mathcal{H}$ es *universal* si $\forall u_1, u_2 \in U$:

$$\left[u_1 \neq u_2 \;\Rightarrow\; \text{card}\{h \in \mathcal{H} | h(u_1) = h(u_2)\} = \frac{\text{card}(\mathcal{H})}{m} \right] \qquad (3)$$

$h : x \mapsto h_x(x)$, $h_x \in \mathcal{H}$ elegida aleatoriamente.

Dispersión universal

Observación 2 *Si $\mathcal{H}$ es universal, n: número de llaves, m: tamaño tabla, entonces $\forall u$, esperanza del número de colisiones donde u aparezca es $\frac{n-1}{m}$.*

$\forall u, v \in U$, $u \neq v$, sean $c_{uv} = \begin{cases} 1 & \text{si } h(u) = h(v), \\ 0 & \text{en otro caso.} \end{cases}$ y $C_u = \sum_{v \neq u} c_{uv}$.

Se tiene, $E(c_{uv}) = \frac{1}{m}$, y $E(C_u) = \sum_{v \neq u} E(c_{uv}) = \frac{n-1}{m}$.

Construcción colección universal $\mathcal{H}$

- Supongamos que m: número primo. Sea $r > 0$. A cada x le asociamos r+1 secuencias de símbolos $\mathbf{x} = (x_r, \ldots, x_1, x_0)$.
- Por ejemplo, si x se representa en binario con k *bits*, entonces x_i: i-ésimo bloque de $\lfloor k/r \rfloor$ *bits*.
- Ahora, sea $\mathbf{a} \in [\![0, m-1]\!]^{r+1} =< a_0, a_1, \ldots, a_r >$ un conjunto de $r+1$ números elegidos aleatoriamente del conjunto $0, 1, \ldots, m-1$
- La función

  $h_{\mathbf{a}} : x \mapsto (\mathbf{x} \cdot \mathbf{a}) \bmod m = (\sum_{i=0}^{r} x_i a_i) \bmod m.$
- Sea $\mathcal{H} = \{h_{\mathbf{a}} | \mathbf{a} \in [\![0, m-1]\!]^{r+1}\}$. $\text{card}(\mathcal{H}) = m^{r+1}$.

Observación 3 $\mathcal{H}$ *es universal.*

$\forall x, y \in U$, sean $\mathbf{x}, \mathbf{y}$ correspondientes vectores. Supongamos $x_0 \neq y_0$. Ya que m es primo, luego $\forall a_1, \ldots, a_r \in [\![0, m-1]\!]$, fijos, $\exists! a_0 \in [\![0, m-1]\!]$ que hace que $h(x) = h(y)$. a_0 es la solución a

$$a_0(x_0 - y_0) = \left(\sum_{i=1}^{r} a_i(x_i - y_i) \right) \bmod m,$$

Ya que m es primo, la cantidad $x_0 - y_0$ diferente de cero tiene un inverso multiplicativo $mod m$, y así hay una solución única para $a_0 mod m$.

Por tanto hay exactamente m^r funciones donde x e y colisionan. Así pues, prob{x e y colisionen}=$\frac{m^r}{m^{r+1}} = \frac{1}{m}$.

Por lo tanto, $\mathcal{H}$ es universal.

Direccionamiento abierto

- Tabla de dispersión hace las veces de almacenamiento.
- Cada vez que llega un nuevo elemento de la tabla dinámica, se *explora* la tabla de dispersión para colocarlo en una posición libre.
- $n \leq m$
- U: llaves, T: tabla de dispersión de longitud m.
- Función de *exploración*, $h : U \times [\![0, m-1]\!] \to [\![0, m-1]\!]$: $\forall u \in U$

$$(h(k,0), h(k,1), \ldots, h(k,m-1)) \text{ permutación de } [\![0, m-1]\!] \quad (4)$$

(u, x): registro de la tabla dinámica, u: llave, x: contenido.

```
IntroducirEnTabDisp(I, (u, x))
{
    i := 0 ; Seguir := Verdadero ;
    while Seguir ∧ (i ≤ m) do {
        j := h(u, i) ;
        if I[j] ≠ nil then
            i++
        else {
            I[j] := (u, x) ; Seguir := Falso
        }
    } ;
    if Seguir then '¡No más cupo en la tabla!'
}
```

```
BúsquedaEnTabDisp(I, (u, x))
{
    i := 0 ; Seguir := Verdadero ;
    while Seguir ∧ (i ≤ m) do {
        j := h(u, i) ;
        if I[j] ≠ (u, x) then i++
        else Seguir := Falso
    } ;
    if Seguir then '¡No aparece (u, x)!'
    else return j
}
```

Explora: Cada consulta al valor de una entrada.

- La operación de inserción introduce nuevos elementos en primeras localidades vacías.
- Dificulta el borrado, o supresión, de elementos:
- Si para suprimir a un elemento se sustituye, en la entrada i, el valor por nil, entonces podría perderse acceso a llaves que al ser insertadas, i estaba ocupada.
- Puede evitarse esto marcando como `Suprimida` a una entrada borrada y, al insertar a un nuevo elemento, considerarlas vacías. Sin embargo, la complejidad del algoritmo de búsqueda variará y ya no dependerá sólo del orden de arribo de elementos.
- Una función de exploración es *uniforme* si con cada llave, las $m!$ permutaciones son igualmente probables según (4).

Exploración Lineal y Cuadrática

Exploración lineal $h_0 : U \to [\![0, m-1]\!]$ de dispersión.
$h^* : (u, i) \mapsto (h_0(u) + i) \bmod m$.

$\forall u \in U$, $(h^*(u,i))_i = (h_0(u) + i) \bmod m)_i$ depende de $h_0(u) \in [\![0, m-1]\!]$. Define a lo más m perm's. h^* no es uniforme.

Exploración cuadrática $h_0 : U \to [\![0, m-1]\!]$ de dispersión.
$h^* : (u, i) \mapsto (h_0(u) + c_1 i + c_2 i^2) \bmod m$.

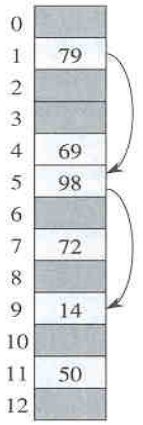
Igualmente, define a lo sumo m perm's. h^* no puede ser uniforme.

Exploración Doble

Exploración doble $h_0, h_1 : U \to [\![0, m-1]\!]$ dos f'nes de dispersión, h_1 tal que $\forall u \in U$, $h_1(u)$ es primo relativo con m.
$h^* : (u, i) \mapsto (h_0(u) + i h_1(u)) \bmod m$.

- Por ejemplo, m potencia de 2 y h_1 tal que sólo asuma valores nones.
- O bien, m primo y h_1 tal que sólo asuma valores estrictamente positivos (y menores que m).
- Define a lo más m^2 perm's. h^* no puede ser uniforme. Mejor desempeño que las exploraciones lineal y cuadrática

Función de Dispersión Doble



Inserción por función de dispersión doble. Se usan la funciones $h_1(k) = k \bmod 13$ y $h_2(k) = 1 + k \bmod 11$

Análisis

$\alpha = \frac{n}{m}$: razón de carga, n registros con tabla de long. m.

$n \leq m$ y $\alpha \leq 1$. $h : U \times [\![0, m-1]\!] \to [\![0, m-1]\!]$ uniforme.

Proposición 1 (Tiempo esperado búsquedas no-exitosas) *Número esperado de pruebas en una búsqueda no exitosa acotado superiormente por* $\frac{1}{1-\alpha}$.

Cada prueba, excepto la última, se topa con entradas ya ocupadas, y la última vacía.

$\forall k \in I\!N$, $p_k = \text{prob}\{\text{exactamente } k \text{ entradas exploradas ya ocupadas}\}$.

$(k > n \Rightarrow p_k = 0)$. No. esperado pruebas: $1 + \sum_{k \geq 0} k p_k$.

Sea $q_k = \text{prob}\{\text{al menos } k \text{ entradas exploradas ya ocupadas}\}$. Entonces, $\sum_{k \geq 0} k p_k = \sum_{k \geq 1} q_k$.

$q_1 = \frac{n}{m} = \alpha$, y, $q_k = \prod_{j=0}^{k-1} \frac{n-j}{m-j} \leq \left(\frac{n}{m}\right)^k = \alpha^k$. Por tanto, $1 + \sum_{k \geq 1} q_k \leq \frac{1}{1-\alpha}$.

Análisis de Inserción

Corolario 1 *Número de pruebas para insertar un elemento es a lo sumo* $\frac{1}{1-\alpha}$.

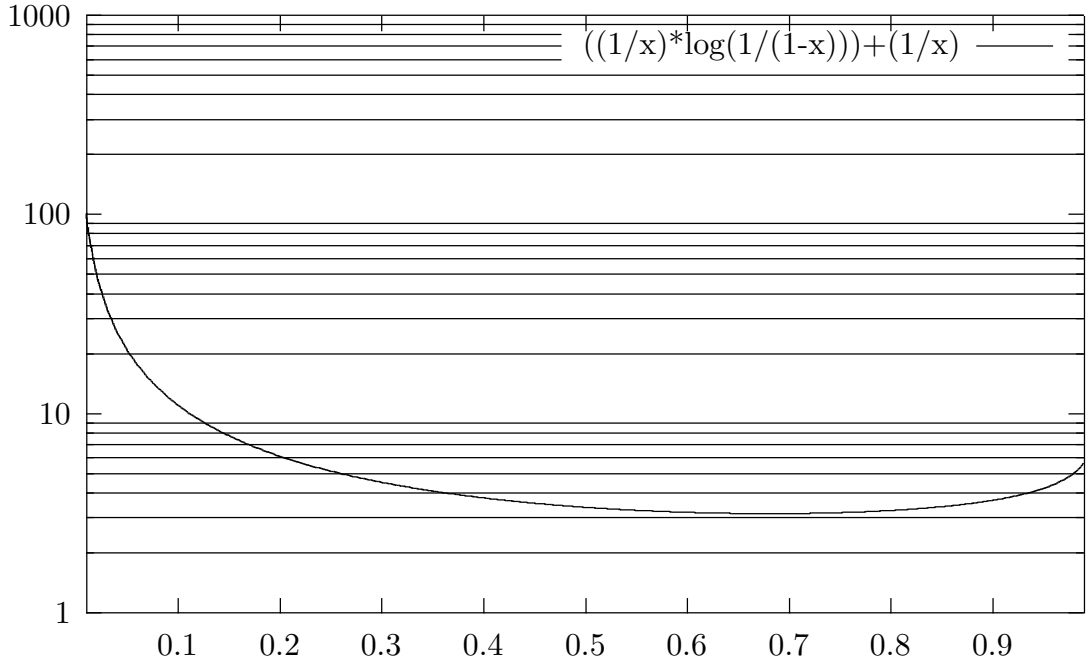
Para realizar una inserción hay que ver que aún hay lugar.

Proposición 2 (Tiempo esperado de búsquedas exitosas) *Supongamos cada llave igualmente probable de aparecer. El número esperado de pruebas en una búsqueda exitosa esta acotado superiormente por* $\frac{1}{\alpha}\left(1+\ln\frac{1}{1-\alpha}\right)$.

La búsqueda de una u requiere tantas pruebas cuantas fueron realizadas cuando se insertó.

Número esperado de entradas exploradas para la $(j+1)$-ésima u_j: $\dfrac{1}{1-\frac{j}{m}} = \dfrac{m}{m-j}$.

Así pues, $\dfrac{1}{n}\displaystyle\sum_{j=0}^{n-1}\frac{m}{m-j} = \frac{m}{n}\sum_{j=0}^{n-1}\frac{1}{m-j} = \frac{1}{\alpha}\left(H_m - H_{m-n}\right)$ donde $H_k = \sum_{j=1}^{k}\frac{1}{j}$.


((1/x)*log(1/(1-x)))+(1/x)
1000
100
10
1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9