

UNIVERSIDAD DE ORIENTE

NÚCLEO DE ANZOÁTEGUI

DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS

ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS

CÁTEDRA: Estructura de Datos

SECCIÓN 01

**Tablas Hash**

Profesor: Integrantes:

Gabriela Veracierta. Br. Luis Agreda C.I: 26.886.935

Br.Gabriel Trujillo C.I 25.244.508

Barcelona, febrero de 2022

1. **Se tiene una aplicación en la que se espera que se manejen 50 elementos. ¿Cuál es el tamaño apropiado de una tabla hash que los almacene?**

Una manera de hallar el tamaño adecuado es dentro de la ecuación donde se representa el factor de carga , la recomienda para elegir alfa es que sea 0.5< <=0.8, y el tamaño de la tabla tiene que ser un número mayor y primo más cercano.

La ecuación de factor de carga es:

Donde:

N=representa la cantidad de elemento en el universo de clave.

M que es la incógnita , hay que reemplazarla por el numero primo más cercano, y de manera que sea lo más cercano a 0.8 ,en este caso 67.

Este requisito se debe que con el valor del factor de carga podemos obtener el número de probabilidad de que una celda esté vacía y así limita el número de intentos de inserción hasta encontrar una celda disponible.

**2. Una tabla hash se ha implementado con exploración lineal, considerando que actualmente el factor de carga es 0.30. ¿Cuáles el número esperado de posiciones de la tabla que se prueban en una búsqueda de una clave, con éxito y sin éxito?**

Para inserciones y búsquedas no exitosas el número de intentos aproximado sería:

**1/2(1 + 1/(1 – λ)**2) .

Su poniendo que

Para búsquedas exitosas sería: 1/2(1 + 1/(1 - λ)), un número menor que el anterior.

Entonces se puede verificar que, el numero esperado de posiciones de la tabla que se prueban en una búsqueda de una clave es sin éxito porque el numero esperado de intentos es mayor al de con éxito.

**3. Para la secuencia de claves: 29, 41, 22, 31, 50, 19, 42, 38; una tabla hash de tamaño 12, y la función hash aritmética modular, mostrar las posiciones de almacenamiento suponiendo la exploración lineal.**

M = 12

Arreglo= {0..11}

H(N) = N mod M

Para la siguientes claves:

1. H(29) = 29 mod 12 = 5 2. H(41) = 41 mod 12 = **5**

3. H(22) = 22 mod 12 = 10 4. H(31) = 31 mod 12 = 7

5. H(50) = 50 mod 12 = 2 6. H(19) = 19 mod 12 = **7**

7. H(42) = 42 mod 12 = **6**  8. H(38) = 38 mod 12 = **2**

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 | 50 |
| 3 | 38 |
| 4 |  |
| 5 | 29 |
| 6 | 41 |
| 7 | 31 |
| 8 | 19 |
| 9 | 42 |
| 10 | 22 |
| 11 |  |

**4. Se va a utilizar como función hash el método mitad del cuadrado. Si el tamaño de la tabla es de 100, encontrar los índices que le corresponden a las claves: 2134, 5231, 2212, 1011.**

M=100 (tamaño de la tabla)

El número de dígitos a extraer depende del rango de dispersión que se quiere obtener. Así, si el rango es de [0…99] se extraen dos dígitos.

Los correspondientes índices para las clavesserian la extracción de dos dígitos de las posiciones 4 y 5 por la derecha dando como resultado.

H(N) = N2

H(2134) =21342 = 4.5**53.**956 = 53 Para 2134 su índice es 53

H(5231) =52312 = 27.3**63**.361 = 63 Para 5231 su índice es 63

H(2212) =22122 = 4.8**92**.944 = 92 Para 2212 su índice es 92

H(1011) =10112 = 1.0**22**.121 = 22 Para 1011 su índice es 22

**5. Para resolución de colisiones se utiliza el método de doble función hash. Siendo la primera función el método aritmético modular y la segunda el método de la multiplicación. Encontrar las posiciones que ocupan los elementos con claves: 14, 31, 62, 26, 39, 44, 45, 22, 15, 16 en una tabla de tamaño 17.**

M = 17

Arreglo = {0...16}

H(14) = 14 mod 17 = 14 no hay colisiones

H(31) = 31 mod 17 = 14 hay colisiones

Metodo de la multiplicacion

H(N) = | M ((N \* R) mod 1) |

Donde R es la inversa de la razón áurea, R = 0.6180334 y donde mod 1 debe ser interpretado como k\*A - | k\*A |

H(31) = | 17 ((31 \* 0,6180334) mod 1) |

H(31) = 2

Se suman los resultados de los métodos

H(31) = 14 + 2 = 16 no hay colisiones

H(62) = 62 mod 17 =11 no hay colisiones

H(26) = 26 mod 17 = 9 no hay colisiones

H(39) = 39 mod 17 = 5 no hay colisiones

H(44) = 44 mod 17 = 10 no hay colisiones

H(45) = 45 mod 17 = 11 hay colisión

H(45) = | 17 (45 \* 0,6180334) mod 1) | = 13

H(45) = 11 + 13 = 24 no hay colisión

Como se desborda, una manera de calcular su posición es volver a sacar el modulo del tamaño de la tabla.

24 mod 17 = 7.

H(45) =7. No hay colision

H(22) = 22 mod 17 = 5 hay colisión

H(22) = | 17 (22 \* 0,6180334) mod 1) | = 10

H(22) = 5 +10 = 15 no hay colisión

H(15) = 15 mod 17 = 15 hay colisión

H(15) = | 17 (15 \* 0,6180334) mod 1) | = 4

H(15) = 15 + 4 = 19

H(15) = 19 mod 17 = 2

H(15) = 2 no hay colisión

H(16) = 16 mod 17 = 16 hay colisión

H(16) = | 17 (16 \* 0,6180334) mod 1) | = 15

H(16) = 16 + 15 = 31 no hay colisión

H(16) = 31 mod 17 = 14 colisionó nuevamente.

Se aplica p 2p´

H(16) = 16 + 2(15) = 46

H(16) = 46 mod 17 = 12 no hay colision.

H(16) = 12.

Quedaría de esta manera la tabla

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 15 |  |  | 39 |  | 45 |  | 26 | 44 | 62 | 16 |  | 14 | 22 | 31 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

**6. En el hashing enlazado la operación de insertar un elemento se ha realizado como primer elemento de la lista. Si la inserción en la lista enlazada se hace de tal forma que esté ordenada respecto la clave. ¿Qué ventajas y desventajas tiene respecto a la eficiencia de las operaciones insertar, buscar y eliminar?**

En el caso de la operacion insertar es una desventaja, ya que en la alternativa que se propone se requiere insertar ordenado en la posición correspondiente, lo cual dependerá del número de colisiones que se produzcan y dejará de ser constante pasaría a un orden mayor.

En las funciones buscar y eliminar, se puede decir que es ventaja en los casos en que la clave que se quiere buscar o eliminar no exista. Sin embargo, siempre se recorre toda la lista, con la nueva alternativa sólo se recorre hasta encontrarlo o encontrar una clave mayor o menor a la que se está buscando.