ESTRUCTURAS DE DATOS

DICCIONARIOS

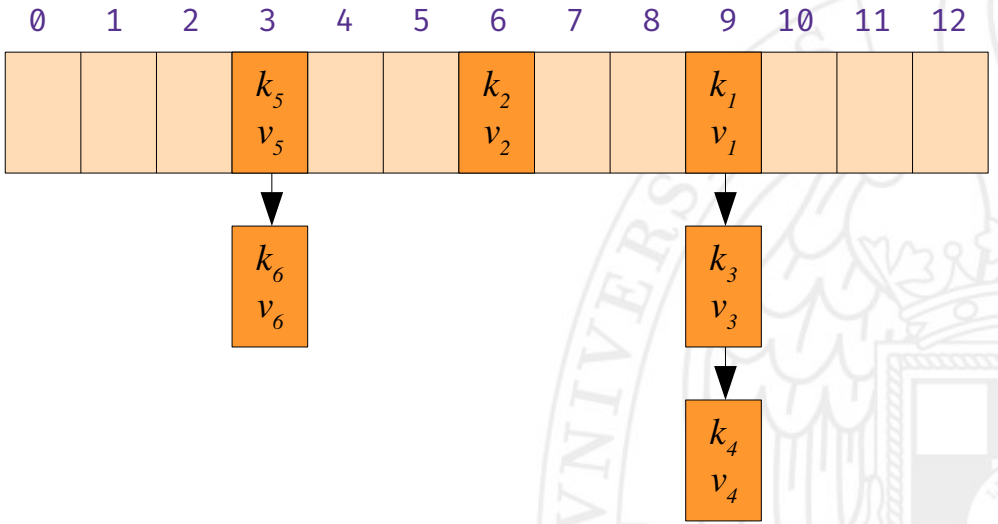
Análisis de coste en tablas *hash*

Manuel Montenegro Montes
Departamento de Sistemas Informáticos y Computación
Facultad de Informática – Universidad Complutense de Madrid

Tablas *hash* abiertas

Recordatorio

- Una tabla *hash* abierta asocia cada cajón con una **lista de entradas**.
- En caso de colisión entre claves, las entradas acaban en la misma lista.



Factor de carga

- El **factor de carga** α de una tabla *hash* es el cociente entre el número de entradas en la tabla y el número de cajones.
- Sean:

  n - número de entradas en la tabla

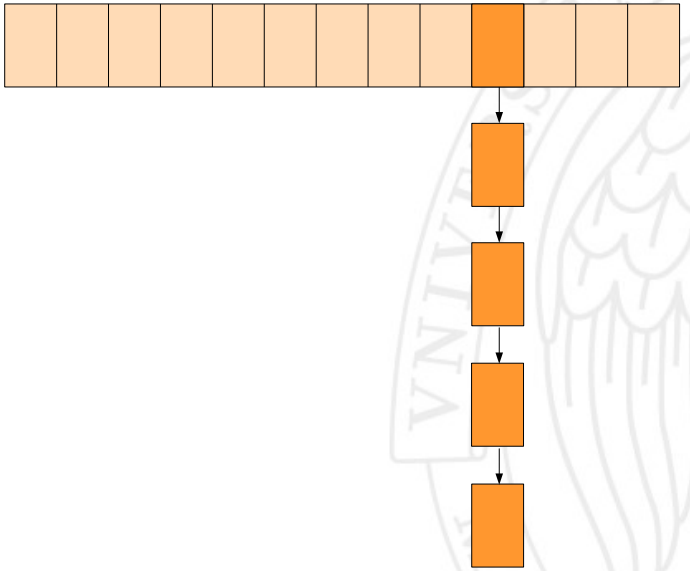
  m – número de cajones

$$\alpha = \frac{n}{m}$$

- Expresaremos el coste de los algoritmos en función del factor de carga.

Dispersión uniforme

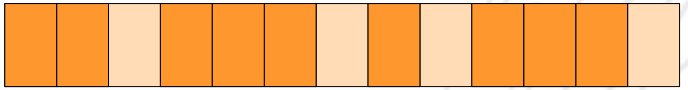
- La eficiencia de una tabla *hash* viene determinada por las propiedades de la función *hash* utilizada.
- Una función *hash* nefasta enviaría todas las claves al mismo cajón.



Dispersión uniforme

Suposición de dispersión uniforme:

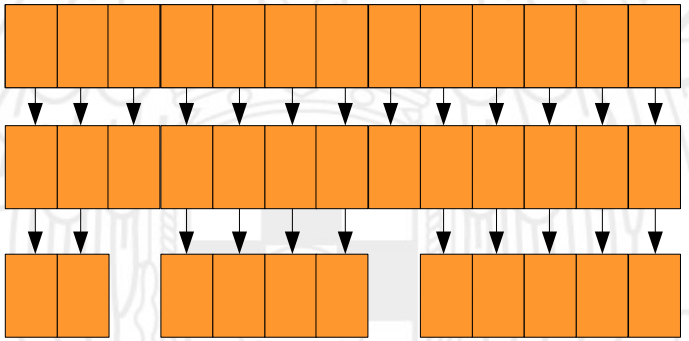
La función *hash* distribuye uniformemente todas las claves a lo largo de los cajones de la tabla.



Longitud media de las listas

- Supongamos que tenemos n elementos y m cajones, y que la propiedad de distribución uniforme se cumple.
- Sea N_i la longitud de la lista del cajón i-ésimo.
- ¿Cuál es el valor promedio de N_i?

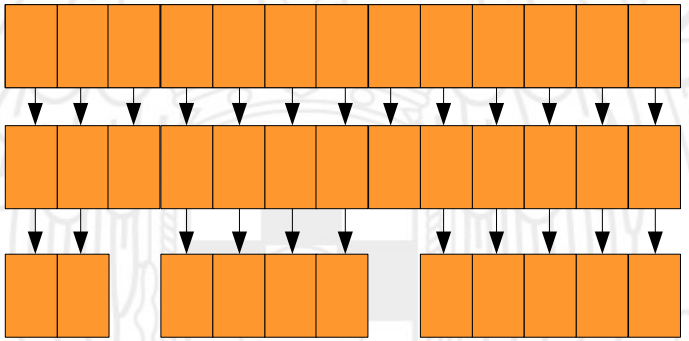
$$E[N_i] = \frac{n}{m} = \alpha$$



Búsqueda de una clave (fallo)

- Supongamos que realizamos una búsqueda de una clave que no se encuentra en la tabla.
- El coste debe recorrer una de las listas en su totalidad.
- Por tanto, el coste medio es proporcional a α.
- Similarmente para la inserción y borrado.

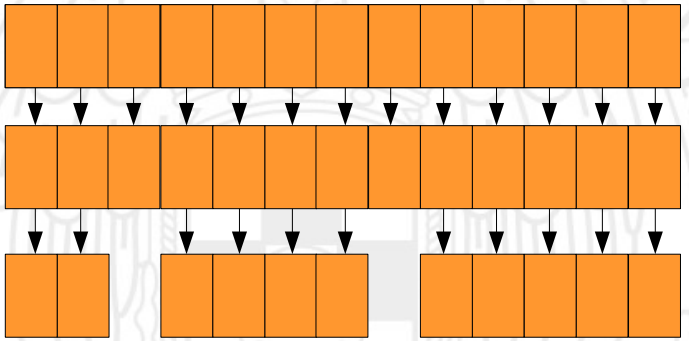
$$O(1+\alpha)$$



Búsqueda de una clave (éxito)

- Supongamos que realizamos una búsqueda de una clave que sí se encuentra en la tabla.
- El número medio de elementos recorridos es $1 + \frac{\alpha}{2} - \frac{\alpha}{2n}$
- Similarmente para la inserción y borrado.

$$1 + \frac{\alpha}{2} - \frac{\alpha}{2n} \in O(1 + \alpha)$$



Conclusión

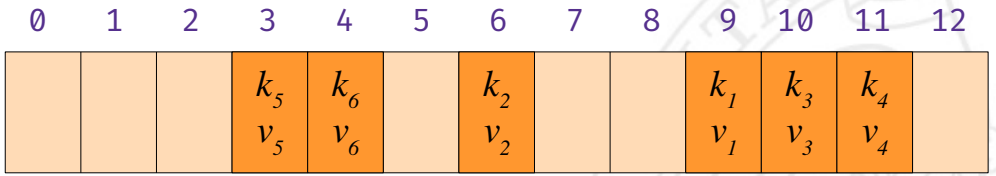
- El coste de todas las operaciones está acotado por α.
- **Si conseguimos mantener α acotado, el coste de las operaciones será constante.**
- ¿Cómo conseguimos mantener α acotado?

  Haciendo que el número de cajones aumente proporcionalmente con el número de entradas → *Tabla dinámicamente redimensionable*.

Tablas *hash* cerradas

Recordatorio

- Una tabla *hash* cerrada contiene, en cada cajón, una única entrada.
- En caso de colisión entre claves, las entradas acaban en cajones distintos según la estrategia de redispersión utilizada.



Factor de carga

- Sean:

  n - número de entradas en la tabla

  m – número de cajones

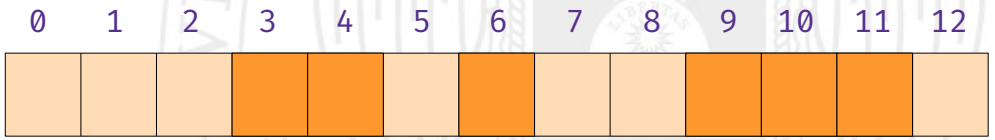
$$\alpha = \frac{n}{m}$$

- En una *tabla hash* cerrada, siempre se cumple que $\alpha \leq 1$.

Búsqueda de una clave (fallo)

- Sea N una variable aleatoria que denota el número de intentos infructuosos hasta que llegamos a un cajón vacío.

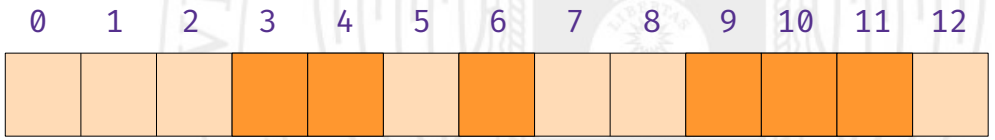
$$P\{N \geq 1\} = \frac{n}{m}$$



Búsqueda de una clave (fallo)

- Sea N una variable aleatoria que denota el número de intentos infructuosos hasta que llegamos a un cajón vacío.

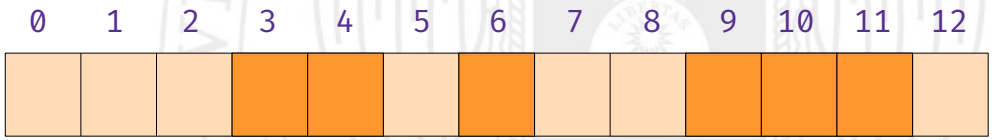
$$P\{N \geq 2\} = \frac{n}{m} * \frac{n-1}{m-1}$$



Búsqueda de una clave (fallo)

- Sea N una variable aleatoria que denota el número de intentos infructuosos hasta que llegamos a un cajón vacío.

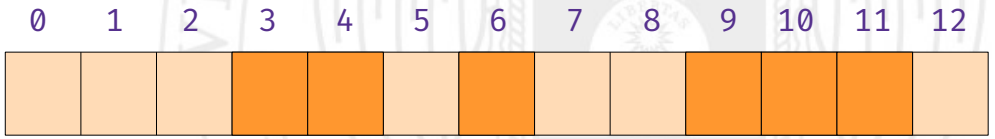
$$P\{N \geq 3\} = \frac{n}{m} * \frac{n-1}{m-1} * \frac{n-2}{m-2}$$



Búsqueda de una clave (fallo)

- Sea N una variable aleatoria que denota el número de intentos infructuosos hasta que llegamos a un cajón vacío.

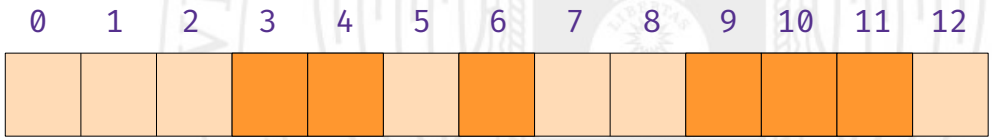
$$P\{N \geq i\} = \frac{n}{m} * \frac{n-1}{m-1} * \frac{n-2}{m-2} * \cdots * \frac{n-i+1}{m-i+1}$$



Búsqueda de una clave (fallo)

- Sea N una variable aleatoria que denota el número de intentos infructuosos hasta que llegamos a un cajón vacío.

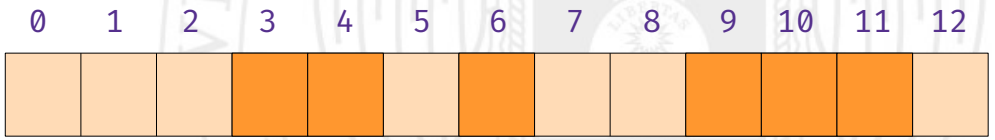
$$P\{N \geq i\} \leq \alpha^i$$



Búsqueda de una clave (fallo)

- Promedio de cajones visitados para buscar una clave:

$$1 + E[N] = 1 + \sum_{i=1}^{\infty} P\{N \geq i\} \leq 1 + \sum_{i=1}^{\infty} \alpha^i = \sum_{i=0}^{\infty} \alpha^i = \frac{1}{1-\alpha}$$



Búsqueda de una clave (fallo)

- Promedio de cajones visitados para buscar una clave:

$$1 + E[N] = 1 + \sum_{i=1}^{\infty} P\{N \geq i\} \leq 1 + \sum_{i=1}^{\infty} \alpha^i = \sum_{i=0}^{\infty} \alpha^i = \frac{1}{1-\alpha}$$

- Si $\alpha = 0.9$, entonces se visitan *10* cajones en el caso medio.
- Si $\alpha = 0.8$, entonces se visitan *5* cajones en el caso medio.
- Si $\alpha = 0.5$, entonces se visitan *2* cajones en el caso medio.

Búsqueda de una clave (éxito)

- Promedio de cajones visitados para buscar una clave:

$$\frac{1}{\alpha} * \ln \frac{1}{1-\alpha}$$

- Si $\alpha = 0.9$, entonces se visitan 2.56 cajones en el caso medio.
- Si $\alpha = 0.8$, entonces se visitan 2 cajones en el caso medio.
- Si $\alpha = 0.5$, entonces se visitan 1.38 cajones en el caso medio.

Conclusión

- **Si conseguimos mantener α inferior a 1, el coste de las operaciones será constante.**
- Con $\alpha \leq 0.8$ se obtienen constantes razonables.
- ¿Cómo conseguimos mantener α constante?
  - Haciendo que el número de cajones aumente proporcionalmente con el número de entradas → *Tabla dinámicamente redimensionable*.

Bibliografía

- T. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein
  Introduction to Algorithms (3ª edición)
  The MIT Press (2009)
  Capítulo 11
- R. Peña
  Diseño de Programas. Formalismo y Abstracción (3ª edición)
  Pearson Educación (2005)
  Sección 8.1.3