# ¿Filtro Bloom o hash de cuco? - algoritmo, hash, filtro

cuál prefieres y por qué?

Ambos pueden usarse para realizar tareas similares, pero tengo curiosidad por ver qué personas han usado en las aplicaciones reales y su razonamiento para hacerlo.

Respuestas

*9 para la respuesta № 1*

Los filtros Bloom y los filtros Cuckoo se usan en situaciones similares, pero hay muchas diferencias debajo que generalmente determinan cuál es una mejor opción.

Los filtros Bloom se usan internamente en la base de datosmotores, notablemente Apache Cassandra. Las razones son, como dicen otros carteles, para reducir el costo de las operaciones lentas. Básicamente, cualquier operación de "si esto no existe o definitivamente no existe" con un alto costo puede usar un filtro Bloom para reducir la cantidad de comprobaciones realizadas.

Otro ejemplo común con el modelo de SaaS de hoySería un servicio REST remoto con un costo por llamada. Cualquier llamada API con una respuesta binaria como "es esta dirección NO VÁLIDA" puede usar un filtro de floración para eliminar más del 90% de las consultas duplicadas. Tenga en cuenta que dado que los filtros Bloom y Cuckoo tienen falsos positivos, NO son útiles para la operación inversa "es esta dirección VÁLIDA"

Importante recordar es que Bloom y Cuckoolos filtros NO tienen falsos negativos. Esto hace que estos filtros sean útiles para comprobaciones como "definitivamente esto no es así o tal vez es correo no deseado", pero no es útil para operaciones en las que los falsos positivos son inaceptables, como verificar los permisos de los usuarios. En este aspecto, pueden conceptualmente considerarse lo opuesto a un caché. Tanto el filtro Bloom como el cuco se usan principalmente para reducir el costo de operaciones costosas con una respuesta booleana, excepto que las memorias caché no tienen falsos positivos y Bloom / Cuckoo no tiene falsos negativos.

Las diferencias notables entre Cuckoo / Bloom incluyen:

* Combinación. Los filtros Bloom se pueden fusionar eficientemente siempre que se creen con los mismos parámetros. Rápido y con poco ancho de banda. Esta es la razón por la que los ve con frecuencia en sistemas distribuidos masivamente, intercambiar filtros Bloom es rápido. Los filtros de cuco no son fácilmente compostables, lo que los hace menos útiles en estas circunstancias.
* Tasa de falsos positivos. Los filtros Cuckoo son más eficientes en cuanto a espacio. Muchos casos de uso para ambas estructuras se centran en redes de bajo nivel. En hardware débil, la eficacia ~ 40% mayor de los filtros Cuckoo para la misma tasa de falsos positivos puede ser importante. La implementación de referencia, en c ++, clasifica los elementos dentro de cada segmento para ahorrar espacio adicional, aprovechando la posición de un elemento dentro de un cubo para almacenar huellas dactilares más pequeñas. Las bibliotecas adicionales que mencionaré más adelante (incluida la mía) no parecen haz esto. Si alguien alguna vez usa mi biblioteca, podría agregarla :).
* Tasa constante de falsos positivos. Los filtros Bloom tienen tasas asintóticamente peores de falsos positivos a medida que superan el tamaño diseñado. Puede seguir insertando elementos para siempre, pero finalmente su tasa de falsos positivos será casi del 100%. Los filtros de cuco, basados ​​en hashing de Cuckoo, tienen una capacidad establecida en la que las inserciones realmente fallarán. La repetición de la inserción de hashes de elementos no aleatorios puede hacer que los filtros Cuckoo fallen su inserción, posiblemente mucho antes de su nivel de llenado diseñado.
* Velocidad. Esto es subjetivo y depende mucho del hardware, pero los filtros Cuckoo generalmente son más rápidos en el caso promedio (según mi experiencia). La mayoría de los diseños de filtro de Bloom ejecutan una función hash dos veces. Al usar funciones hash seguras especialmente, esto puede ser una gran desventaja en comparación con los filtros Cuckoo que solo insertan elementos hash una vez. El código que he visto utiliza varias funciones de hash para los filtros Bloom y Cuckoo. Google Bloom de Guava usa Murmur3, muchas otras implementaciones usan SHA1 u otra cosa. Si las colisiones hash se pueden explotar para su caso de uso, asegúrese de que la biblioteca utilice un hash seguro. Es importante saber que los filtros Bloom tardan aproximadamente un tiempo constante en insertarse, mientras que los filtros Cuckoo tienen un caso PROMEDIO de tiempo constante. A medida que los filtros Cuckoo alcanzan un porcentaje de capacidad, las velocidades de inserción disminuyen considerablemente. Incluso entonces, solo se ralentiza la velocidad de inserción, todas las demás operaciones son tiempo promedio constante.
* Flexibilidad. Los filtros Bloom solo admiten inserción y contienen. Los filtros Cuckoo también son compatibles con la eliminación y el conteo limitado. En el diseño de referencia, los filtros Cuckoo pueden determinar cuántas veces se insertó un artículo, hasta 7 veces. Los filtros Bloom solo pueden determinar si-no. Los filtros Cuckoo también son compatibles con la eliminación de elementos insertados, un gran positivo en muchos casos de uso en comparación con Bloom. Cuando se utilizan filtros Bloom, es bastante normal recrear el filtro desde cero cuando está "lleno" (la tasa estimada de falsos positivos supera el umbral) ya que no se pueden eliminar elementos antiguos. Tenga en cuenta que la reconstrucción del filtro aún ocurre con los filtros Cuckoo cuando se inserta comience a fallar, por lo que dependiendo del caso de uso, esto podría ser discutible. En ciertas situaciones, los filtros Cuckoo son más útiles ya que puede eliminar elementos para mantenerse dentro de los límites del filtro en lugar de reconstruir.
* Apoyo. Los filtros de cuco son bibliotecas nuevas y estables para muchos idiomas simplemente no existen.

La mayor ventaja de los filtros Bloom es quetienen un soporte de biblioteca más maduro en la mayoría de los idiomas. La matemática detrás de los filtros Bloom también es mejor entendida por los científicos. La mayoría de las características de los filtros Cuckoo han sido determinadas empíricamente, mientras que los filtros Bloom tienen una base numérica sólida. Esto excluye los filtros Cuckoo para sistemas críticos y en tiempo real que deben tener verificación de su rendimiento, aunque la evidencia experimental muestra que los filtros Cuckoo funcionan mejor en la mayoría de las circunstancias.

Shameless Plug: soy el desarrollador de una biblioteca de filtros Cuckoo para Java. [CuckooFilter4J](https://github.com/MGunlogson/CuckooFilter4J/) . Le falta el semi-tipo de cubo utilizado en elpapel, por lo que la eficiencia del espacio es algo menor que la implementación de referencia. En el archivo Léame del proyecto, tengo enlaces a otras implementaciones de las que tengo conocimiento. La estructura que es mejor depende de su caso de uso, pero principalmente de si existe una implementación sólida de filtro Cuckoo para su idioma.

Definitivamente deberías echarle un vistazo a la fuenteantes de usar un filtro Cuckoo / Bloom en producción. Leí varias librerías antes de escribir las mías ... muchas de ellas tenían límites de tamaño silenciosos debido a arreglos subyacentes de 32 bits o problemas obvios de rendimiento. La mayoría tenía cero pruebas. La implementación de Google Guava Bloom tuvo la mejor calidad de código y pruebas (y admite límites de matriz de 64 bits). Las únicas deficiencias con Bloom de Guava es que no tiene una opción para usar una función de hash segura y no es " t multihilo.

En un sistema de producción, es posible que deseemulti-threading para la velocidad. La respuesta para Bloom de Guava es hacer un filtro diferente para cada hilo y combinarlos de vez en cuando. Como los filtros Cuckoo no se pueden combinar, agregué el uso simultáneo a mi biblioteca de filtros Cuckoo. El otro soy consciente de que no son seguros o no son concurrentes.

*8 para la respuesta № 2*

¿Qué prefieres, vino o queso?

UN filtro de floración es para cuando tienes espacio limitado, alto costo de consultay consultas en su mayoría negativas.  
En ese caso, un filtro de floración con 8 bits por tecla y 4 funciones hash te dio 2.5% tasa de falsos positivos; usted procesa consultas casi 40 veces más rápido que antes, a costa de 1 byte por clave.

Por otro lado, si alguno de los las condiciones previas no son válidas, un tabla hash que actúa como un cachétiene sentido, aunque obviamente tomará una mucho más de un byte por entrada :-)

Incluso puede omitir los casos de borde duro de hash cuco si es un caché. Eso también hace que los problemas de aumento de tamaño de tablas de hash cuco (o cualquier cosa que no sea hash lineal) discutible.

*5 para la respuesta № 3*

Filtro de cuco.

"Filtro de cuco: Prácticamente mejor que Bloom". Bin Fan, David Andersen, Michael Kaminsky, Michael Mitzenmacher CoNext 2014. <http://dx.doi.org/10.1145/2674005.2674994>

De uno de los autores " [Blog](http://mybiasedcoin.blogspot.com/2014/10/cuckoo-filters.html):

Déjame describir un filtro de cuco y algunos delo que está en el papel para usted. Si desea evitar una discusión técnica, todo lo que necesita saber es que para conjuntos de tamaño razonablemente grande, para la misma tasa de falsos positivos que un filtro Bloom correspondiente, los filtros de cuco usan menos espacio que Bloom filtros, son más rápidos en las búsquedas (pero más lentos en las inserciones / construcciones) y, sorprendentemente, también permiten eliminar las claves (lo que los filtros Bloom no pueden hacer). Si desea ver el código, incluso hay una [repositorio github](https://github.com/efficient/cuckoofilter) para ti con código para filtros de cuco.

*2 para la respuesta № 4*

Prefiero el hash del cuco. Desconfío de los falsos positivos que pueden aparecer con los filtros de floración en factores de relleno más altos.  
Hemos utilizado hashing de cuco en una aplicación donde teníamos tablas hash muy grandes y estábamos teniendo problemas de presión de memoria. Por favor mira mi biblioteca de eCollections en [http://codeplex.com/ecollections](http://www.codeplex.com/ecollections) para la implementación de una variante de hash de cuco.

Saludos cordiales,

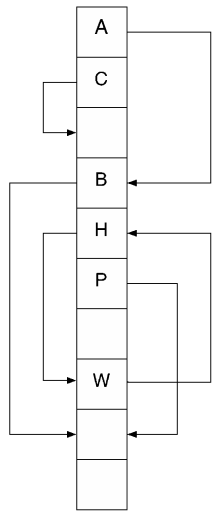
*0 para la respuesta № 5*

Si puedo tolerar los falsos positivos y el espacio es crítico, uso un filtro Bloom porque ocupa menos espacio. De lo contrario, uso un hash.

# Hash de cuco

De Wikipedia, la enciclopedia libre

[Saltar a la navegación](https://en.wikipedia.org/wiki/Cuckoo_hashing#mw-head)[Saltar a la búsqueda](https://en.wikipedia.org/wiki/Cuckoo_hashing#p-search)

[](https://en.wikipedia.org/wiki/File:Cuckoo.svg)

Ejemplo de hashing de cuco. Las flechas muestran la ubicación alternativa de cada tecla. Se insertaría un nuevo artículo en la ubicación de A moviendo A a su ubicación alternativa, actualmente ocupada por B, y moviendo a B a su ubicación alternativa que está actualmente vacante. La inserción de un nuevo elemento en la ubicación de H no tendría éxito: como H es parte de un ciclo (junto con W), el nuevo elemento sería expulsado de nuevo.

**El hash de cuco** es un esquema de [programación de computadora](https://en.wikipedia.org/wiki/Computer_programming) para resolver [colisiones hash](https://en.wikipedia.org/wiki/Hash_collision) de valores de [funciones hash](https://en.wikipedia.org/wiki/Hash_function) en una [tabla](https://en.wikipedia.org/wiki/Hash_table), con el [peor](https://en.wikipedia.org/wiki/Worst_case_analysis) tiempo de búsqueda [constante](https://en.wikipedia.org/wiki/Constant_time) . El nombre se deriva del comportamiento de algunas especies de [cuco](https://en.wikipedia.org/wiki/Cuckoo) , donde el polluelo cuco empuja a los otros huevos o crías fuera del nido cuando eclosiona; análogamente, insertar una nueva clave en una tabla de hash de cuco puede empujar una clave anterior a una ubicación diferente en la tabla.



## Contenido

* [1Historia](https://en.wikipedia.org/wiki/Cuckoo_hashing#History)
* [2operación](https://en.wikipedia.org/wiki/Cuckoo_hashing#Operation)
* [3Teoría](https://en.wikipedia.org/wiki/Cuckoo_hashing#Theory)
* [4Práctica](https://en.wikipedia.org/wiki/Cuckoo_hashing#Practice)
* [5Ejemplo](https://en.wikipedia.org/wiki/Cuckoo_hashing#Example)
  + [5.1Ciclo](https://en.wikipedia.org/wiki/Cuckoo_hashing#Cycle)
* [6variaciones](https://en.wikipedia.org/wiki/Cuckoo_hashing#Variations)
* [7Comparación con estructuras relacionadas](https://en.wikipedia.org/wiki/Cuckoo_hashing#Comparison_with_related_structures)
* [8Vea también](https://en.wikipedia.org/wiki/Cuckoo_hashing#See_also)
* [9referencias](https://en.wikipedia.org/wiki/Cuckoo_hashing#References)
* [10Enlaces externos](https://en.wikipedia.org/wiki/Cuckoo_hashing#External_links)
  + [10.1Ejemplos](https://en.wikipedia.org/wiki/Cuckoo_hashing#Examples)

## Historia [ [editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit&section=1)]

El hashing del cuco fue descrito por primera vez por [Rasmus Pagh](https://en.wikipedia.org/w/index.php?title=Rasmus_Pagh&action=edit&redlink=1) y [Flemming Friche Rodler](https://en.wikipedia.org/w/index.php?title=Flemming_Friche_Rodler&action=edit&redlink=1) en 2001. [[1]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-Cuckoo-1)

## Operación [ [editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit&section=2)]

El hash de cuco es una forma de [direccionamiento abierto](https://en.wikipedia.org/wiki/Open_addressing) en el que cada celda no vacía de una [tabla hash](https://en.wikipedia.org/wiki/Hash_table) contiene una [clave](https://en.wikipedia.org/wiki/Unique_key) o [par clave-valor](https://en.wikipedia.org/wiki/Attribute%E2%80%93value_pair) . Se usa una [función hash](https://en.wikipedia.org/wiki/Hash_function) para determinar la ubicación de cada tecla, y su presencia en la tabla (o el valor asociado a ella) se puede encontrar examinando esa celda de la tabla. Sin embargo, el direccionamiento abierto sufre [colisiones](https://en.wikipedia.org/wiki/Collision_(computer_science)), que ocurre cuando se mapea más de una clave en la misma celda. La idea básica del hash de cuco es resolver colisiones usando dos funciones hash en lugar de solo una. Esto proporciona dos ubicaciones posibles en la tabla hash para cada clave. En una de las variantes comúnmente utilizadas del algoritmo, la tabla hash se divide en dos tablas más pequeñas de igual tamaño, y cada función hash proporciona un índice en una de estas dos tablas. También es posible que ambas funciones hash proporcionen índices en una sola tabla.

La búsqueda requiere la inspección de solo dos ubicaciones en la tabla hash, lo que lleva tiempo constante en el peor de los casos ( *consulte la*[notación Big O](https://en.wikipedia.org/wiki/Big_O_notation) ). Esto está en contraste con muchos otros algoritmos de tabla hash, que pueden no tener un peor caso constante enlazado en el momento de hacer una búsqueda. Las eliminaciones también pueden realizarse cerrando la celda que contiene una clave, en el peor caso constante de tiempo, más simplemente que algunos otros esquemas, como el [sondeo lineal](https://en.wikipedia.org/wiki/Linear_probing) .

Cuando se inserta una nueva clave, y una de sus dos celdas está vacía, puede colocarse en esa celda. Sin embargo, cuando ambas celdas ya están llenas, será necesario mover otras claves a sus segundas ubicaciones (o volver a sus primeras ubicaciones) para dejar espacio para la nueva clave. Se usa un [algoritmo codicioso](https://en.wikipedia.org/wiki/Greedy_algorithm) : la nueva clave se inserta en una de sus dos ubicaciones posibles, "expulsando", es decir, desplazando, cualquier clave que ya resida en esta ubicación. Esta clave desplazada luego se inserta en su ubicación alternativa, nuevamente expulsando cualquier tecla que pueda residir allí. El proceso continúa de la misma manera hasta que se encuentra una posición vacía, completando el algoritmo. Sin embargo, es posible que este proceso de inserción falle al ingresar un [ciclo infinito](https://en.wikipedia.org/wiki/Infinite_loop)o al encontrar una cadena muy larga (más larga que un umbral preestablecido que es [logarítmico](https://en.wikipedia.org/wiki/Logarithm) en el tamaño de la tabla). En este caso, la tabla hash se reconstruye [in situ](https://en.wikipedia.org/wiki/In-place_algorithm) utilizando nuevas [funciones hash](https://en.wikipedia.org/wiki/Hash_function) :

No es necesario asignar nuevas tablas para el reajuste: simplemente podemos ejecutar las tablas para eliminar y llevar a cabo el procedimiento de inserción habitual en todas las teclas que no se encuentran en la posición prevista en la tabla.

-  Pagh y Rodler, "Cuckoo Hashing" [[1]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-Cuckoo-1)

## Teoría [ [editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit&section=3)]

Las inserciones tienen éxito en tiempo constante esperado, [[1]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-Cuckoo-1) incluso considerando la posibilidad de tener que reconstruir la tabla, siempre que el número de claves se mantenga por debajo de la mitad de la capacidad de la tabla hash, es decir, el [factor de carga](https://en.wikipedia.org/wiki/Load_factor_(computer_science)) está por debajo del 50%.

Un método para probar esto usa la teoría de [gráficos aleatorios](https://en.wikipedia.org/wiki/Random_graph) : uno puede formar un [gráfico no dirigido](https://en.wikipedia.org/wiki/Undirected_graph) llamado "gráfico de cuco" que tiene un vértice para cada ubicación de tabla hash, y un borde para cada valor hash, con los puntos finales del borde siendo el dos posibles ubicaciones del valor. Luego, el algoritmo de inserción codicioso para agregar un conjunto de valores a una tabla de hash cuco tiene éxito si y solo si el gráfico de cuco para este conjunto de valores es un [pseudoforesto](https://en.wikipedia.org/wiki/Pseudoforest) , un gráfico con un ciclo como máximo en cada uno de sus [componentes conectados](https://en.wikipedia.org/wiki/Connected_component_(graph_theory)) . Cualquier subgráfico inducido por vértices con más aristas que vértices corresponde a un conjunto de claves para las cuales hay un número insuficiente de ranuras en la tabla de almohadilla. Cuando la función de hash se elige al azar, el gráfico de cuco es[gráfico al azar](https://en.wikipedia.org/wiki/Random_graph) en el [modelo Erdős-Rényi](https://en.wikipedia.org/wiki/Erd%C5%91s%E2%80%93R%C3%A9nyi_model) . Con alta probabilidad, para un gráfico aleatorio en el que la relación entre el número de aristas y el número de vértices está limitada por debajo de 1/2, el gráfico es pseudoforestal y el algoritmo de hash de cuco tiene éxito al colocar todas las claves. Además, la misma teoría también demuestra que el tamaño esperado de un [componente conectado](https://en.wikipedia.org/wiki/Connected_component_(graph_theory)) del gráfico de cuco es pequeño, asegurando que cada inserción tome un tiempo esperado constante. [[2]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-2)

## Practica [ [editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit&section=4)]

En la práctica, el hash Cuckoo es aproximadamente un 20-30% más lento que el [sondeo lineal,](https://en.wikipedia.org/wiki/Linear_probing) que es el más rápido de los enfoques comunes. [[1]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-Cuckoo-1) La razón es que el hash de cuco a menudo causa dos fallas de caché por búsqueda, para verificar las dos ubicaciones donde se puede almacenar una clave, mientras que el sondeo lineal generalmente causa solo una falta de caché por búsqueda. Sin embargo, debido a sus peores garantías de caso en el tiempo de búsqueda, el hash de cuco puede ser valioso cuando se requieren [tasas de respuesta en tiempo real](https://en.wikipedia.org/wiki/Real-time_computing) .

## Ejemplo [ [editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit&section=5)]

Se dan las siguientes funciones hash:

{\ displaystyle h \ left (k \ right) = k \ mod 11}  
{\ displaystyle h '\ left (k \ right) = \ left \ lfloor {\ frac {k} {11}} \ right \ rfloor \ mod 11}

|  |  |  |
| --- | --- | --- |
| **k** | **h (k)** | **h '(k)** |
| 20 | 9 | 1 |
| 50 | 6 | 4 |
| 53 | 9 | 4 |
| 75 | 9 | 6 |
| 100 | 1 | 9 |
| 67 | 1 | 6 |
| 105 | 6 | 9 |
| 3 | 3 | 0 |
| 36 | 3 | 3 |
| 39 | 6 | 3 |

Las columnas de las dos tablas siguientes muestran el estado de las tablas hash a lo largo del tiempo a medida que se insertan los elementos.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1. tabla para h (k)** | | | | | | | | | | | | | | |
|  | 20 | 50 | 53 | 75 | 100 | 67 | 105 | | 3 | | 36 | | 39 | |
| 0 |  |  |  |  |  |  |  | |  | |  | |  | |
| 1 |  |  |  |  | 100 | 67 | 67 | | 67 | | 67 | | 100 | |
| 2 |  |  |  |  |  |  |  | |  | |  | |  | |
| 3 |  |  |  |  |  |  |  | | 3 | | 36 | | 36 | |
| 4 |  |  |  |  |  |  |  | |  | |  | |  | |
| 5 |  |  |  |  |  |  |  | |  | |  | |  | |
| 6 |  | 50 | 50 | 50 | 50 | 50 | 105 | | 105 | | 105 | | 50 | |
| 7 |  |  |  |  |  |  |  | |  | |  | |  | |
| 8 |  |  |  |  |  |  |  | |  | |  | |  | |
| 9 | 20 | 20 | 53 | 75 | 75 | 75 | 53 | | 53 | | 53 | | 75 | |
| **2. tabla para h '(k)** | | | | | | | | | | | | | | |
|  | 20 | 50 | 53 | 75 | 100 | 67 | | 105 | | 3 | | 36 | | 39 |
| 0 |  |  |  |  |  |  | |  | |  | | 3 | | 3 |
| 1 |  |  | 20 | 20 | 20 | 20 | | 20 | | 20 | | 20 | | 20 |
| 2 |  |  |  |  |  |  | |  | |  | |  | |  |
| 3 |  |  |  |  |  |  | |  | |  | |  | | 39 |
| 4 |  |  |  | 53 | 53 | 53 | | 50 | | 50 | | 50 | | 53 |
| 5 |  |  |  |  |  |  | |  | |  | |  | |  |
| 6 |  |  |  |  |  |  | | 75 | | 75 | | 75 | | 67 |
| 7 |  |  |  |  |  |  | |  | |  | |  | |  |
| 8 |  |  |  |  |  |  | |  | |  | |  | |  |
| 9 |  |  |  |  |  | 100 | | 100 | | 100 | | 100 | | 105 |

### Ciclo [ [editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit&section=6)]

Si ahora desea insertar el elemento 6, entra en un ciclo. En la última fila de la tabla encontramos la misma situación inicial que al principio otra vez.

{\ displaystyle h \ left (6 \ right) = 6 \ mod 11 = 6}  
{\ displaystyle h '\ left (6 \ right) = \ left \ lfloor {\ frac {6} {11}} \ right \ rfloor \ mod 11 = 0}

|  |  |
| --- | --- |
| **tabla 1** | **Tabla 2** |
| 6 reemplaza a 50 en la celda 6 | 50 reemplaza a 53 en la celda 4 |
| 53 reemplaza a 75 en la celda 9 | 75 reemplaza a 67 en la celda 6 |
| 67 reemplaza 100 en la celda 1 | 100 reemplaza a 105 en la celda 9 |
| 105 reemplaza 6 en la celda 6 | 6 reemplaza 3 en la celda 0 |
| 3 reemplaza 36 en la celda 3 | 36 reemplaza a 39 en la celda 3 |
| 39 reemplaza a 105 en la celda 6 | 105 reemplaza 100 en la celda 9 |
| 100 reemplaza 67 en la celda 1 | 67 reemplaza a 75 en la celda 6 |
| 75 reemplaza a 53 en la celda 9 | 53 reemplaza a 50 en la celda 4 |
| 50 reemplaza a 39 en la celda 6 | 39 reemplaza a 36 en la celda 3 |
| 36 reemplaza a 3 en la celda 3 | 3 reemplaza 6 en la celda 0 |
| 6 reemplaza a 50 en la celda 6 | 50 reemplaza a 53 en la celda 4 |

## Variaciones [ [editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit&section=7)]

Se han estudiado varias variaciones de hash de cuco, principalmente con el objetivo de mejorar su uso de espacio al aumentar el [factor de carga](https://en.wikipedia.org/wiki/Load_factor_(computer_science)) que puede tolerar a un número mayor que el umbral del 50% del algoritmo básico. Algunos de estos métodos también se pueden usar para reducir la tasa de fallas del hash de cuco, lo que provoca que las reconstrucciones de la estructura de datos sean mucho menos frecuentes.

Se puede esperar que las generalizaciones de hash de cuco que utilizan más de dos funciones de hash alternativas utilicen una mayor parte de la capacidad de la tabla de hash de manera eficiente a la vez que se sacrifica la búsqueda y la velocidad de inserción. Usar solo tres funciones hash aumenta la carga al 91%. [[3]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-mitzenmacher2009survey-3) Otra generalización del hash de cuco, llamada *hash de cuco bloqueado,* consiste en utilizar más de una clave por cubo. El uso de solo 2 llaves por cubo permite un factor de carga superior al 80%. [[4]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-4)

Otra variación del hash de cuco que se ha estudiado es el *hash de cuco con un alijo* . El alijo, en esta estructura de datos, es una matriz de un número constante de claves, que se usa para almacenar claves que no pueden insertarse con éxito en la tabla principal de hash de la estructura. Esta modificación reduce la tasa de fallas del hash de cuco a una función de polinomio inverso con un exponente que puede hacerse arbitrariamente grande al aumentar el tamaño del escondite. Sin embargo, los stashes más grandes también significan búsquedas más lentas para las claves que no están presentes o están en el escondite. Un alijo se puede usar en combinación con más de dos funciones hash o con hash de cuco bloqueado para lograr tanto factores de alta carga como tasas de falla pequeñas. [[5]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-5)El análisis del hash de cuco con un stash se extiende a funciones hash prácticas, no solo al modelo de función hash aleatorio comúnmente utilizado en el análisis teórico de hash. [[6]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-6)

Algunas personas recomiendan una generalización simplificada de hash de cuco denominada [caché asociada asimétrica](https://en.wikipedia.org/wiki/CPU_cache#Two-way_skewed_associative_cache) en algunos [cachés de CPU](https://en.wikipedia.org/wiki/CPU_cache) . [[7]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-7)

Otra variación de una tabla hash de cuco, llamada filtro de cuco, reemplaza las teclas almacenadas de una tabla de hash cuco con huellas dactilares mucho más cortas, calculadas aplicando otra función hash a las teclas. Para permitir que estas huellas dactilares se muevan dentro del filtro de cuco, sin conocer las claves de las que proceden, las dos ubicaciones de cada huella digital se pueden calcular entre sí mediante una operación [exclusiva a nivel de](https://en.wikipedia.org/wiki/Exclusive_or) bit [o](https://en.wikipedia.org/wiki/Exclusive_or) con la huella dactilar, o con un hash de la huella digital Esta estructura de datos forma una estructura de datos de membresía de conjunto aproximado con las mismas propiedades que un [filtro de Bloom](https://en.wikipedia.org/wiki/Bloom_filter) : puede almacenar los miembros de un conjunto de claves y probar si una clave de consulta es un miembro, con alguna posibilidad de [falsos positivos.](https://en.wikipedia.org/wiki/False_positive)(consultas que se informaron incorrectamente como parte del conjunto) pero no [falsos negativos](https://en.wikipedia.org/wiki/False_negative) . Sin embargo, mejora en un filtro Bloom en múltiples aspectos: su uso de memoria es más pequeño por un factor constante, tiene mejor [localidad de referencia](https://en.wikipedia.org/wiki/Locality_of_reference) y (a diferencia de los filtros Bloom) permite una eliminación rápida de elementos establecidos sin penalización adicional de almacenamiento. [[8]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-8)

## Comparación con estructuras relacionadas [ [editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit&section=8)]

Un estudio de Zukowski et al. [[9]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-9) ha demostrado que el hash de cuco es mucho más rápido que el [hashing encadenado](https://en.wikipedia.org/wiki/Separate_chaining) para tablas de hash pequeñas y con [memoria caché](https://en.wikipedia.org/wiki/CPU_cache) en procesadores modernos. Kenneth Ross [[10]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-10) ha demostrado que las versiones compartidas de hash cuco (variantes que usan cubos que contienen más de una clave) son más rápidas que los métodos convencionales también para tablas hash grandes, cuando la utilización del espacio es alta. El rendimiento de la tabla de hash de cuco recortada fue investigado más por Askitis, [[11]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-11) con su rendimiento comparado con esquemas de hash alternativos.

Una encuesta de [Mitzenmacher](https://en.wikipedia.org/wiki/Michael_Mitzenmacher)[[3]](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_note-mitzenmacher2009survey-3) presenta problemas abiertos relacionados con el hash de cuco a partir de 2009.

## Ver también [ [editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit&section=9)]

* [Hashing perfecto](https://en.wikipedia.org/wiki/Perfect_hashing)
* [Doble hash](https://en.wikipedia.org/wiki/Double_hashing)
* [Exploración cuadrática](https://en.wikipedia.org/wiki/Quadratic_probing)
* [Hashing Rayuela](https://en.wikipedia.org/wiki/Hopscotch_hashing)

## Referencias [ [editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit&section=10)]

* 1. ^ [Saltar a:***a***](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-Cuckoo_1-0)[***b***](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-Cuckoo_1-1)[***c***](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-Cuckoo_1-2)[***d***](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-Cuckoo_1-3) Pagh, Rasmus; Rodler, Flemming Friche (2001). "Cuckoo Hashing". [*Algoritmos - ESA 2001*](http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.25.4189) . Notas de la conferencia en informática. **2161** . pp. 121-133. [*doi*](https://en.wikipedia.org/wiki/Digital_object_identifier) : [*10.1007 / 3-540-44676-1\_10*](https://doi.org/10.1007/3-540-44676-1_10) . [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*978-3-540-42493-2*](https://en.wikipedia.org/wiki/Special:BookSources/978-3-540-42493-2) .
  2. [**Salta hacia arriba^**](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-2) Kutzelnigg, Reinhard (2006). [*Gráficos aleatorios bipartitos y hash de cuco*](https://www.dmtcs.org/dmtcs-ojs/index.php/proceedings/article/viewFile/dmAG0133/1710.pdf) (PDF) . Cuarto Coloquio sobre Matemáticas e Informática. Matemáticas Discretas y Computación Teórica. **AG** . pp. 403-406
  3. ^ [Saltar a:***a***](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-mitzenmacher2009survey_3-0)[***b***](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-mitzenmacher2009survey_3-1) [*Mitzenmacher, Michael*](https://en.wikipedia.org/wiki/Michael_Mitzenmacher) (2009-09-09). [*"Algunas preguntas abiertas relacionadas con Cuckoo Hashing | Procedimientos de ESA 2009"*](http://www.eecs.harvard.edu/~michaelm/postscripts/esa2009.pdf)(PDF) *. Consultado 2010-11-10* .
  4. [**Jump up^**](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-4) Dietzfelbinger, Martin; Weidling, Christoph (2007), "Asignación equilibrada y diccionarios con contenedores de tamaño constante fuertemente empaquetados", Theoret. Comput. Sci. ,**380** (1 - 2):[*47 - 68*](https://doi.org/10.1016/j.tcs.2007.02.054) ,[*doi*](https://en.wikipedia.org/wiki/Digital_object_identifier) :[*10.1016 / j.tcs.2007.02.054*](https://doi.org/10.1016/j.tcs.2007.02.054) ,[*MR*](https://en.wikipedia.org/wiki/Mathematical_Reviews) [*2330641*](https://www.ams.org/mathscinet-getitem?mr=2330641) .
  5. [**Jump up^**](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-5) Kirsch, Adam; Mitzenmacher, Michael D .; Wieder, Udi (2010), "Hashing más robusto: hashing de cuco con un alijo", SIAM J. Comput. ,**39** (4):[*1543 - 1561*](https://doi.org/10.1137/080728743) ,[*doi*](https://en.wikipedia.org/wiki/Digital_object_identifier) :[*10.1137 / 080728743*](https://doi.org/10.1137/080728743) ,[*MR*](https://en.wikipedia.org/wiki/Mathematical_Reviews) [*2580539*](https://www.ams.org/mathscinet-getitem?mr=2580539) .
  6. [**Jump up^**](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-6) Aumüller, Martin; Dietzfelbinger, Martin; Woelfel, Philipp (2014), "Familias de hash explícitas y eficientes son suficientes para el hash del cuco con un alijo", Algorithmica ,**70**(3): 428-456,[*arXiv*](https://en.wikipedia.org/wiki/ArXiv) :[*1204.4431*](https://arxiv.org/abs/1204.4431)*Libremente accesible* ,[*doi*](https://en.wikipedia.org/wiki/Digital_object_identifier) :[*10.1007 / s00453-013-9840-x*](https://doi.org/10.1007/s00453-013-9840-x) ,[*MR*](https://en.wikipedia.org/wiki/Mathematical_Reviews) [*3247374*](https://www.ams.org/mathscinet-getitem?mr=3247374) .
  7. [**^ Arriba**](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-7)["Micro-arquitectura"](http://www.irisa.fr/caps/PROJECTS/Architecture/).
  8. [**Jump up^**](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-8) Fan, Bin; Andersen, Dave G .; Kaminsky, Michael; [*Mitzenmacher, Michael D.*](https://en.wikipedia.org/wiki/Michael_Mitzenmacher) (2014), "Filtro de cuco: Prácticamente mejor que Bloom", Proc. 10º ACM Int. Conf. Experimentos y tecnologías de redes emergentes (CoNEXT '14) , pp. 75-88,[*doi*](https://en.wikipedia.org/wiki/Digital_object_identifier) :[*10.1145 / 2674005.2674994*](https://doi.org/10.1145/2674005.2674994)
  9. [**Jump up^**](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-9) Zukowski, Marcin; Heman, Sandor; Boncz, Peter (junio de 2006). [*"Hashing de Arquitectura-Conciencia"*](http://www.cs.cmu.edu/~damon2006/pdf/zukowski06archconscioushashing.pdf)(PDF) . Actas del Taller internacional sobre gestión de datos en hardware nuevo (DaMoN)*. Obtenido el 2008-10-16* .
  10. [**Salta hacia arriba^**](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-10) Ross, Kenneth (2006-11-08). [*"Sondas Hash eficientes en procesadores modernos"*](http://domino.research.ibm.com/library/cyberdig.nsf/papers/DF54E3545C82E8A585257222006FD9A2/$File/rc24100.pdf)(PDF) . Informe de investigación de IBM RC24100. RC24100*. Obtenido el 2008-10-16* .
  11. [**Salta hacia arriba^**](https://en.wikipedia.org/wiki/Cuckoo_hashing#cite_ref-11) Askitis, Nicolás (2009). [*Tablas Hash rápidas y compactas para llaves enteras*](http://crpit.com/confpapers/CRPITV91Askitis.pdf) (PDF) . Actas de la 32ª Conferencia Australasia de Ciencias de la Computación (ACSC 2009) . **91** . pp. 113-122. [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*978-1-920682-72-9*](https://en.wikipedia.org/wiki/Special:BookSources/978-1-920682-72-9) .

## Enlaces externos [ [editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit&section=11)]

* [Una alternativa fresca y práctica a las tablas hash tradicionales](http://www.ru.is/faculty/ulfar/CuckooHash.pdf) , U. Erlingsson, M. Manasse, F. Mcsherry, 2006.
* [Cuckoo Hashing para estudiantes de pregrado, 2006](http://www.it-c.dk/people/pagh/papers/cuckoo-undergrad.pdf) , R. Pagh, 2006.
* [Cuckoo Hashing, teoría y práctica](http://mybiasedcoin.blogspot.com/2007/06/cuckoo-hashing-theory-and-practice-part.html) (Parte 1, [Parte 2](http://mybiasedcoin.blogspot.com/2007/06/cuckoo-hashing-theory-and-practice-part_15.html) y [Parte 3](http://mybiasedcoin.blogspot.com/2007/06/cuckoo-hashing-theory-and-practice-part_19.html) ), Michael Mitzenmacher, 2007.
* Naor, Moni; Segev, Gil; Wieder, Udi (2008). [*"Historia independiente de Cuckoo Hashing"*](http://www.wisdom.weizmann.ac.il/~naor/PAPERS/cuckoo_hi_abs.html) . Coloquio internacional sobre autómatas, lenguajes y programación (ICALP) . Reykjavik, Islandia *. Obtenido el 2008-07-21* .
* [Mejoras algorítmicas para cuclillas hash concurrentes rápidas](http://www.cs.princeton.edu/~mfreed/docs/cuckoo-eurosys14.pdf) , X. Li, D. Andersen, M. Kaminsky, M. Freedman. EuroSys 2014.

### Ejemplos [ [editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit&section=12)]

* [Simulacro de cuco simultáneo de alto rendimiento escrito en C ++](https://github.com/efficient/libcuckoo)
* [Mapa hash Cuco escrito en C ++](http://sourceforge.net/projects/cuckoo-cpp/)
* [Generador de tablas de cuco estático para C / C ++](http://www.theiling.de/projects/lookuptable.html)
* [Tabla de hash Cuckoo escrita en Haskell](http://hackage.haskell.org/packages/archive/hashtables/latest/doc/html/Data-HashTable-ST-Cuckoo.html)
* [Hashing de cuco para Go](https://github.com/salviati/cuckoo)

[Categorías](https://en.wikipedia.org/wiki/Help:Category) :

* [Algoritmos de búsqueda](https://en.wikipedia.org/wiki/Category:Search_algorithms)
* [Hashing](https://en.wikipedia.org/wiki/Category:Hashing)

## Menú de Navegación

* Sin iniciar sesión
* [Hablar](https://en.wikipedia.org/wiki/Special:MyTalk)
* [Contribuciones](https://en.wikipedia.org/wiki/Special:MyContributions)
* [Crear una cuenta](https://en.wikipedia.org/w/index.php?title=Special:CreateAccount&returnto=Cuckoo+hashing)
* [Iniciar sesión](https://en.wikipedia.org/w/index.php?title=Special:UserLogin&returnto=Cuckoo+hashing)
* [Artículo](https://en.wikipedia.org/wiki/Cuckoo_hashing)
* [Hablar](https://en.wikipedia.org/wiki/Talk:Cuckoo_hashing)
* [Leer](https://en.wikipedia.org/wiki/Cuckoo_hashing)
* [Editar](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=edit)
* [Ver historial](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=history)

### Buscar

Principio del formulario



Final del formulario

* [Pagina principal](https://en.wikipedia.org/wiki/Main_Page)
* [Contenido](https://en.wikipedia.org/wiki/Portal:Contents)
* [Contenido destacado](https://en.wikipedia.org/wiki/Portal:Featured_content)
* [Eventos actuales](https://en.wikipedia.org/wiki/Portal:Current_events)
* [Artículo al azar](https://en.wikipedia.org/wiki/Special:Random)
* [Donar a Wikipedia](https://donate.wikimedia.org/wiki/Special:FundraiserRedirector?utm_source=donate&utm_medium=sidebar&utm_campaign=C13_en.wikipedia.org&uselang=en)
* [Tienda Wikipedia](https://shop.wikimedia.org/)

### Interacción

* [Ayuda](https://en.wikipedia.org/wiki/Help:Contents)
* [Acerca de Wikipedia](https://en.wikipedia.org/wiki/Wikipedia:About)
* [Portal de la comunidad](https://en.wikipedia.org/wiki/Wikipedia:Community_portal)
* [Cambios recientes](https://en.wikipedia.org/wiki/Special:RecentChanges)
* [Pagina de contacto](https://en.wikipedia.org/wiki/Wikipedia:Contact_us)

### Herramientas

* [Qué enlaces aquí](https://en.wikipedia.org/wiki/Special:WhatLinksHere/Cuckoo_hashing)
* [Cambios relacionados](https://en.wikipedia.org/wiki/Special:RecentChangesLinked/Cuckoo_hashing)
* [Subir archivo](https://en.wikipedia.org/wiki/Wikipedia:File_Upload_Wizard)
* [Páginas especiales](https://en.wikipedia.org/wiki/Special:SpecialPages)
* [Enlace Permanente](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&oldid=855079568)
* [Información de la página](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&action=info)
* [Artículo de Wikidata](https://www.wikidata.org/wiki/Special:EntityPage/Q1584911)
* [Citar esta página](https://en.wikipedia.org/w/index.php?title=Special:CiteThisPage&page=Cuckoo_hashing&id=855079568)

### Imprimir / exportar

* [Crea un libro](https://en.wikipedia.org/w/index.php?title=Special:Book&bookcmd=book_creator&referer=Cuckoo+hashing)
* [Descargar como PDF](https://en.wikipedia.org/w/index.php?title=Special:ElectronPdf&page=Cuckoo+hashing&action=show-download-screen)
* [Versión imprimible](https://en.wikipedia.org/w/index.php?title=Cuckoo_hashing&printable=yes)

### Idiomas

* [Čeština](https://cs.wikipedia.org/wiki/Kuka%C4%8D%C4%8D%C3%AD_ha%C5%A1ov%C3%A1n%C3%AD)
* [Deutsch](https://de.wikipedia.org/wiki/Kuckucks-Hashing)
* [English](https://es.wikipedia.org/wiki/Cuckoo_hashing)
* [עברית](https://he.wikipedia.org/wiki/%D7%92%D7%99%D7%91%D7%95%D7%91_%D7%A7%D7%95%D7%A7%D7%99%D7%99%D7%94)
* [Lietuvių](https://lt.wikipedia.org/wiki/Gegut%C4%97s_mai%C5%A1a)
* [Polski](https://pl.wikipedia.org/wiki/Tablica_mieszaj%C4%85ca#Haszowanie_kuku.C5.82cze)
* [Русский](https://ru.wikipedia.org/wiki/%D0%9A%D1%83%D0%BA%D1%83%D1%88%D0%BA%D0%B8%D0%BD%D0%BE_%D1%85%D0%B5%D1%88%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)
* [Српски / srpski](https://sr.wikipedia.org/wiki/%D0%9A%D1%83%D0%BA%D0%B0%D0%B2%D0%B8%D1%87%D1%98%D0%B5_%D1%85%D0%B5%D1%88%D0%B8%D1%80%D0%B0%D1%9A%D0%B5)

[Editar enlaces](https://www.wikidata.org/wiki/Special:EntityPage/Q1584911#sitelinks-wikipedia)

* Esta página fue editada por última vez el 15 de agosto de 2018, a las 20:01  (UTC) .
* El texto está disponible bajo la [licencia Creative Commons Attribution-ShareAlike](https://en.wikipedia.org/wiki/Wikipedia:Text_of_Creative_Commons_Attribution-ShareAlike_3.0_Unported_License) ; Se pueden aplicar términos adicionales. Al usar este sitio, usted acepta los [Términos de uso](https://wikimediafoundation.org/wiki/Terms_of_Use) y [la Política de privacidad](https://wikimediafoundation.org/wiki/Privacy_policy) . Wikipedia® es una marca registrada de [Wikimedia Foundation, Inc.](https://www.wikimediafoundation.org/) , una organización sin fines de lucro.
* [Política de privacidad](https://foundation.wikimedia.org/wiki/Privacy_policy)
* [Acerca de Wikipedia](https://en.wikipedia.org/wiki/Wikipedia:About)
* [Descargo de responsabilidad](https://en.wikipedia.org/wiki/Wikipedia:General_disclaimer)
* [Contacto Wikipedia](https://en.wikipedia.org/wiki/Wikipedia:Contact_us)
* [Desarrolladores](https://www.mediawiki.org/wiki/Special:MyLanguage/How_to_contribute)
* [Declaración de cookies](https://foundation.wikimedia.org/wiki/Cookie_statement)
* [Vista móvil](https://en.m.wikipedia.org/w/index.php?title=Cuckoo_hashing&mobileaction=toggle_view_mobile)

v