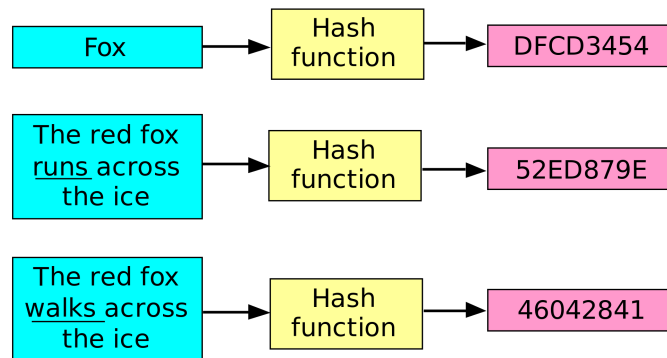


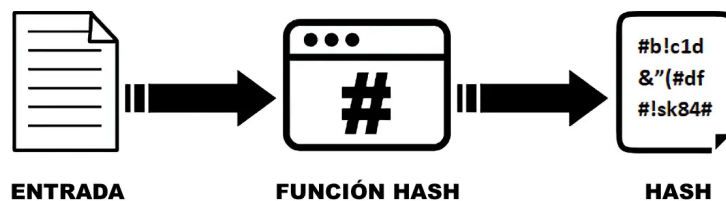
Función, código y tabla hash

Una **función hash** H es una función computable mediante un algoritmo, de manera que H mapea un conjunto de datos de longitud variable U en otro conjunto de datos de longitud fija V :

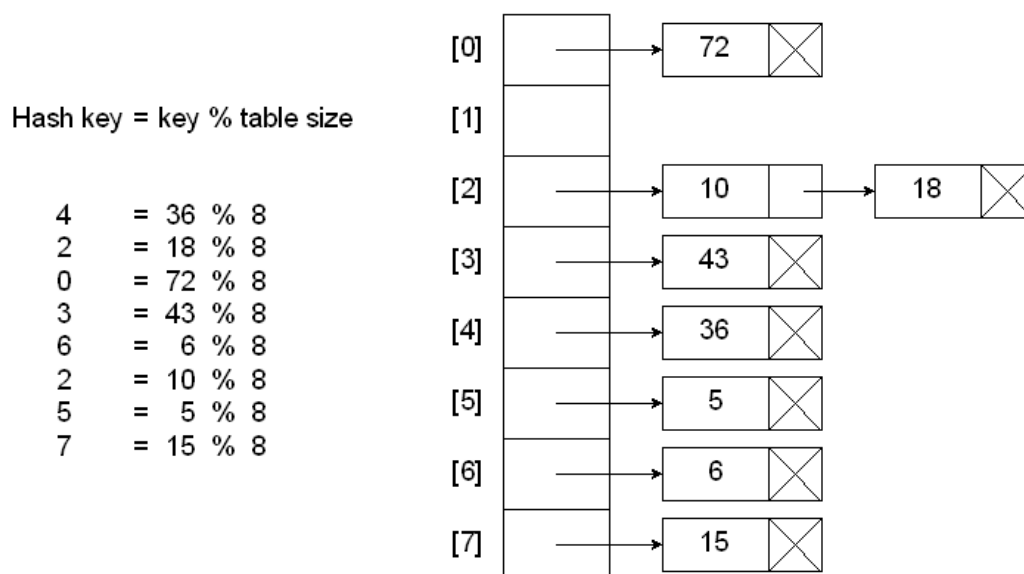
$$H: U \rightarrow V$$



Cada elemento de V generado por H a partir de un elemento de U se denomina **código hash**.



Asignando a la posición i de un array el elemento de entrada U_i cuyo código hash toma el valor i , para todo U_i de U , habremos construido una **tabla de hash**¹.

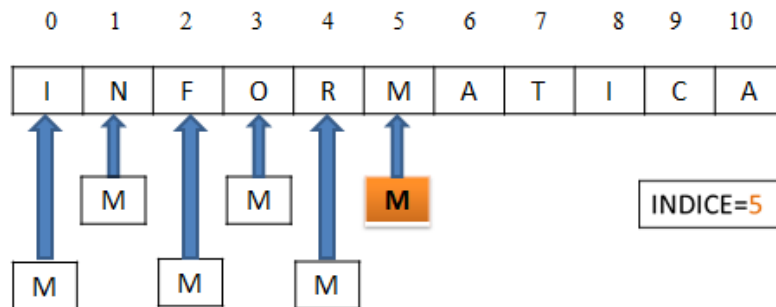


¹ Si dos o más elementos dan lugar al mismo código hash, se produce una colisión de hash, que en el caso de la implementación hashCode de Java, será solventada por una lista enlazada.

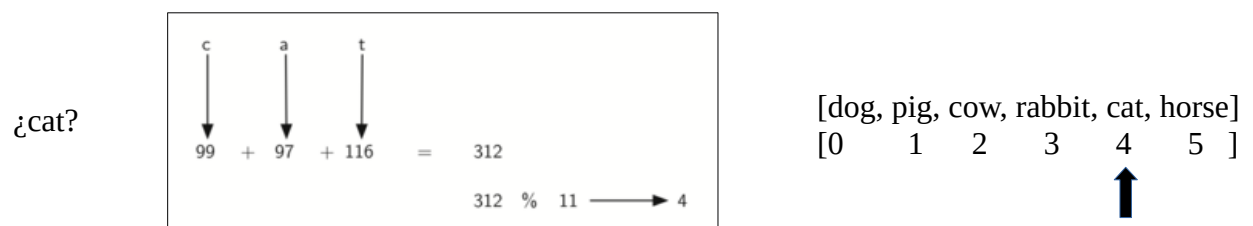
Necesidad del hash

La necesidad de este tipo de estructura de datos se encuentra en la dificultad para llevar a cabo determinadas operaciones simples sobre colecciones de elementos, que en ocasiones se vuelven del todo ineficientes.

Por ejemplo, la simple búsqueda secuencial de un elemento determinado en un array requiere recorrer una cantidad de elementos tan grande como el total de elementos anteriores a la ocurrencia.



Realizando la búsqueda sobre una tabla de hash, podemos determinar la posición del elemento sin recorrer la tabla. Sólo es necesario aplicar la función de hash al elemento que queremos buscar. El código hash obtenido debe corresponderse con la posición en la cual, si existe, debe ubicarse el elemento buscado.



Método hashCode en Java

El método hashCode se encuentra implementado en la clase Object y obtiene el código hash asociado a una determinada instancia².

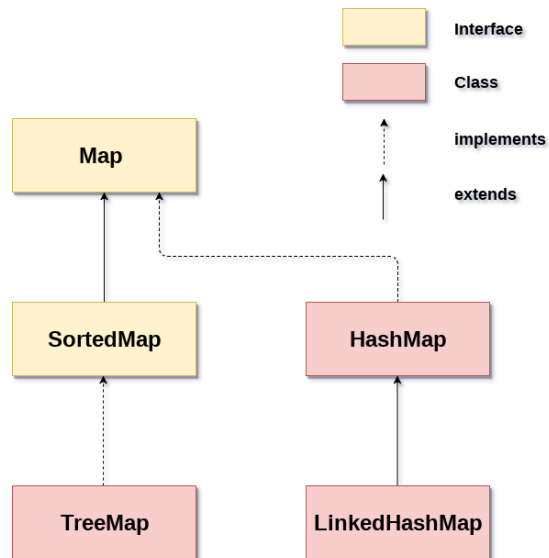


Toda estructura de datos basada en tablas de hash hará uso habitual del método hashCode.

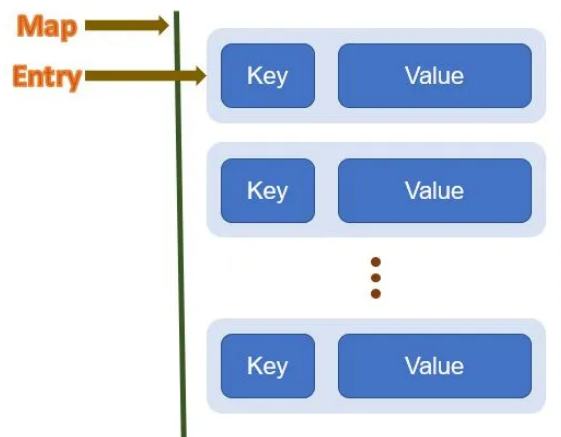
² Nótese que dos instancias de una misma clase pueden poseer el mismo código hash reescribiendo el método hashCode() de la manera adecuada.

Map y HashMap en Java

La interface Map es una estructura de datos en Java que almacena sus elementos en pares {clave:valor} donde cada clave es única. La clase HashMap implementa la interface Map haciendo uso de tablas de hash.



Para funcionar, la clase HashMap hace uso de una clase interna Entry formada por un objeto llave y un objeto valor.



Es precisamente el objeto llave el que pasa por un proceso de hashing, creando su código hash e insertando el Entry completo en una posición determinada dentro de la tabla hash de la implementación hashMap.

Debe notarse que, dado el proceso de construcción, una estructura de datos con implementación hashMap no asegura el orden de inserción de los elementos.