**Hashing functions**

1. Zobrist hashing
2. Aplicaciones

Juegos de mesa.

1. Ventajas

* Puede ser implementado en segundos
* Tiene propiedades teóricos y es generalmente conveniente

1. Desventajas

* Usa mucha memoria.
* Se necesita 8KB para poder pasar todas las cadenas de 4 bytes a valores de 64 bits.

1. Jenkins Hash Function
2. Aplicaciones

La implementación estándar del lenguaje de programación Perl incluye el hash uno a la vez de Jenkins o una variante reforzada del mismo, junto con SipHash, y utiliza el hash uno a la vez de Jenkins de forma predeterminada.

1. Ventajas

* Comportamiento útil para criptografía.

1. Desventajas

* Puede ejecutarse de manera muy lenta.

1. MD5
2. Aplicaciones

* Los servidores de archivos a menudo proporcionan una suma de verificación MD5 precalculada para los archivos así pues, que un usuario pueda comparar la suma de verificación del archivo descargado.
* La mayoría de los sistemas operativos basados ​​en Unix incluyen utilidades de suma MD5 en sus paquetes de distribución.
* Los usuarios de Windows pueden usar la función PowerShell incluida "Get-FileHash", instalar una utilidad de Microsoft o usar aplicaciones de terceros.

1. Ventajas

* Trabaja de manera muy rápida y eficaz.

1. Desventajas

* Es muy propensa a ataques de extensión de longitud.

1. Pearson hashing
2. Aplicaciones

* Útil para implementar Hash tables o códigos de verificación de integridad de datos.

1. Ventajas

* Extremadamente sencillo.
* Se ejecuta rápidamente en procesadores con recursos limitados.
* No existe una clase de entradas simple para las colisiones sean especialmente probables

1. Desventajas

* La tabla de búsqueda de 256 bytes sugerida puede ser excesivamente grande para un microcontrolador pequeño con un tamaño de memoria de programa del orden de cientos de bytes.

1. Fowler–Noll–Vo hash function
2. Aplicaciones

* Diseñado para el uso rápido de la tabla hash y la suma de comprobación

1. Ventajas

* Es sumamente sencillo de implementar.

1. Desventajas

* Si el valor de hash fuera cero en cualquier punto durante el cálculo, y el siguiente byte hash también fuera cero, el hash no cambiaría.