Estructuras de Datos

Hash

Motivación

Supongamos que necesitamos almacenar *n* números enteros, sabiendo de antemano que dichos números se encuentran en un rango conocido por ejemplo:

$$0, \ldots, k-1$$

Para resolver este problema, basta con crear un arreglo de tamaño *k* y marcar con valor `true` o `1` o lo que sea, los casilleros del arreglo cuyo índice sea igual al valor del elemento a almacenar. De esta forma determinamos que el elemento está presente.

Motivación

Un ejemplo de esto sería:

Elementos = {1, 4, 7}

0	1	2	3	4	5	6	7
false	true	false	false	true	false	false	true

Podemos ver que con esta estructura de datos el costo de búsqueda, inserción y eliminación es $O(1)$.

Motivación

Este enfoque tiene dos grandes problemas, si pensamos en el hecho que puedo replicar esta idea para cualquier tipo de dato que se quiera almacenar:

1. El valor de *k* puede ser muy grande, y por lo tanto no habría lugar en memoria para almacenar el arreglo completo.

   Pensemos, por ejemplo, en todos los posibles nombres de personas.

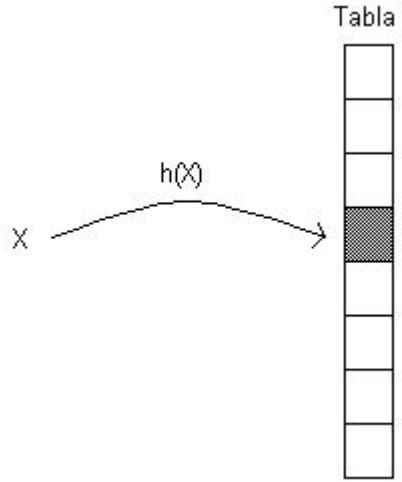
2. Los datos a almacenar pueden ser pocos, con lo cual se estaría desperdiciando espacio de memoria ya que estaríamos pidiendo lugar para todos los datos posibles.

Definiciones - Función de Hash

Nuestro objetivo es poder construir un intervalo de números menor. Para esto vamos a usar una función *h*, denominada *función de hash*.
Esta función es tal que: dado un elemento *X* perteneciente a nuestro universo de datos esperados, en nuestro ejemplo sería un número en el rango [0, ..., k-1], su valor de retorno, es decir *h(X)*, es un número en el rango [0, ..., m-1]. Siendo `m<<k` (m es significativamente más chico que k).
En este caso, se marca el casillero, cuyo índice es *h(X)*, para indicar que el elemento *X* pertenece al conjunto de elementos dados.

Definiciones - Función de Hash

Fíjense que estamos transformando nuestro espacio de valores posibles de *[0, ..., k-1]* *a [0, ..., m-1]*, siendo `m<<k` (`m` es significativamente más chico que k).
Esta estructura de datos es conocida como *tabla hash*.



Al *h(X)* se lo llama clave o key.

Definiciones - Función de Hash

La función *h* debe distribuir los valores lo más uniformemente posible dentro de la tabla, es decir, tendría que ser igualmente probable que salga `0`, `1`, `2`, … o `m-1`.

Dado que tenemos `m` posibles valores, la probabilidad de que la clave obtenida sea `z` *(*donde `z` está en [0, ..., m-1]*)* a partir de un `X` es `1/m`. Esto lo notamos como:

`Pr(h(X)=z)= 1/m` para todo `z` en [0, ..., m-1].

Definiciones - Función de Hash

En general, el valor `X` se puede interpretar como un número entero, y las funciones *h(X)* genéricamente son de la forma:

$$h(X) = (c\ X \bmod p) \bmod m$$

donde *c* es una constante, *p* es un número primo y *m* es el tamaño de la tabla de hashing.

Distintos valores para estos parámetros producen distintas funciones de hash.

Definiciones - Colisiones

El problema que tiene este enfoque es que dos elementos pueden tener la misma clave, es decir, siendo X1 distinto de X2, h(X1) = h(X2).

A este problema se lo denomina Colisiones.

Para ilustrar esto veamos el siguiente ejemplo.

Definiciones - Colisiones

¿Cuál es el número *n* mínimo de personas que es necesario reunir en una sala para que la probabilidad que dos de ella tengan su cumpleaños en el mismo día sea mayor que 1/2?

Hagamos el siguiente razonamiento:

- La primera persona que consideremos puede cumplir años cualquier día, es decir, tiene 365 posibilidades.
- La segunda tiene 365 días menos el del cumpleaño de la persona anterior, es decir: 365-1 posibilidades.
- La tercer persona tiene 365 días menos los días de las 2 personas anteriores, es decir, 365-2.
- La k-ésima persona tiene 365 días menos los días de las k-1 personas anteriores, es decir, 365-(k-1) o, lo que es lo mismo, 365-k+1.

Definiciones - Colisiones

Es decir entonces que la cantidad de posibilidades para n personas sería:

$$365(365-1)(365-2)\ldots\ (365-n+1)$$

Estas serían las posibles fechas de cumpleaños de n personas sin que coincidan en un día. La probabilidad de esto sería dividiendo este valor por los casos posibles, es decir, que cada persona cumpla en cualquiera de los 365 días. Si llamamos d_n a esta probabilidad entonces tenemos que:

$$d_n = \left(\frac{365}{365}\right)\left(\frac{364}{365}\right)\left(\frac{363}{365}\right)\cdots\left(\frac{365-n+1}{365}\right)$$

Definiciones - Colisiones

Responder nuestra pregunta: ¿Cuál es el número *n* mínimo de personas que es necesario reunir en una sala para que la probabilidad que dos de ella tengan su cumpleaños en el mismo día sea mayor que 1/2?

Es lo mismo que: ¿Cuál es el mínimo *n* tal que $d_n < 1/2$?

Respuesta: $n = 23 \Rightarrow d_n = 0.4927$. Notemos que 23 es bastante más chico que 365.

Esto quiere decir que con una pequeña fracción del conjunto de elementos es posible tener colisiones con alta probabilidad.

En la próxima presentación veremos cómo tratar este problema.