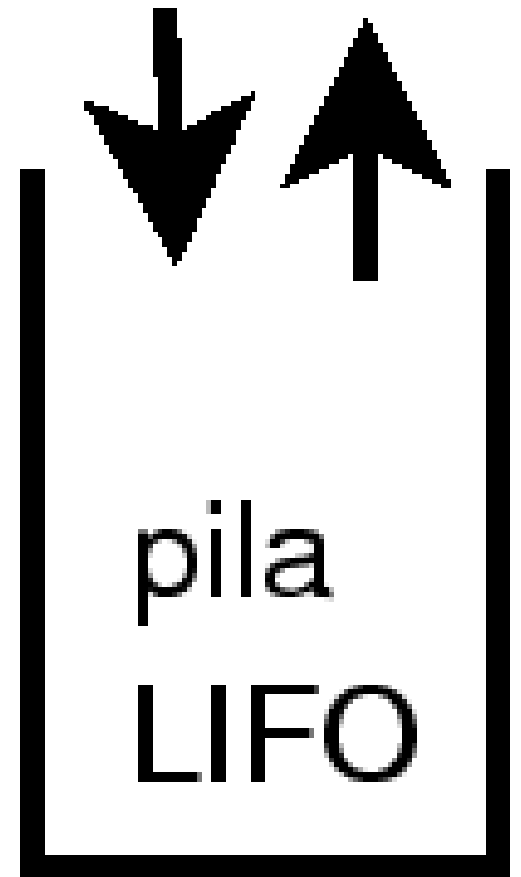


Notas

(Lo más importante a recordar. Lo más elemental)

Pila

- **LIFO (Last In, First Out):** “El último en entrar es el primero en salir”.
- **Solo se puede acceder al elemento en el tope de la pila.**
- **Operaciones:** Push, pop, top.
- (la pila NO dinámica tiene un límite de almacenamiento, la dinámica no).



Cola

- **FIFO (First In, First Out):** “El primero en entrar es el primero en salir”.
- **Los elementos se agregan al final y se eliminan desde el frente.**
- **Operaciones:** Enqueue, dequeue, front, rear.
- (la cola NO dinámica tiene un límite de almacenamiento, la dinámica no).



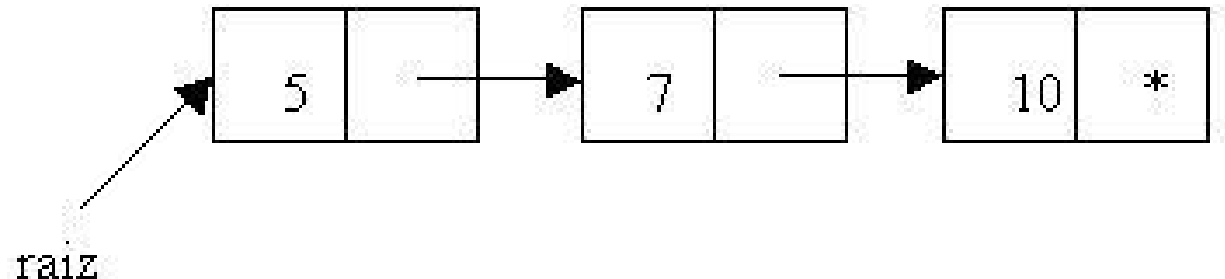
Malloc (en C)

- (memory allocation).
- asignar memoria en el heap de forma dinámica.
- La memoria no se libera automáticamente, es necesario usar `free()`.
- `malloc()` no inicializa la memoria, solo la reserva.
- Si se necesita memoria inicializada a cero, se debe usar `calloc()`.

Lista ligada (simple)

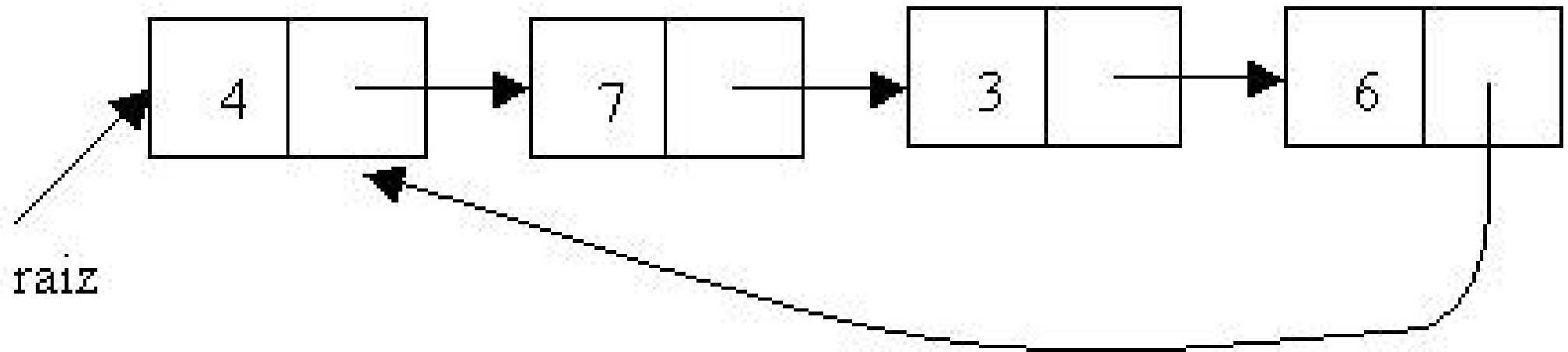
Cada nodo de la lista contiene dos partes:

- Un **dato** (información que almacena el nodo)
 - Un **puntero** que apunta al siguiente nodo en la lista.
-
- **Operaciones:** Inserción, eliminación, recorrido.
 - El primer nodo de la lista se llama **raíz**, y se utiliza un puntero para acceder al primer nodo.
 - Los nodos subsiguientes están enlazados mediante punteros, lo que forma la estructura de la lista.



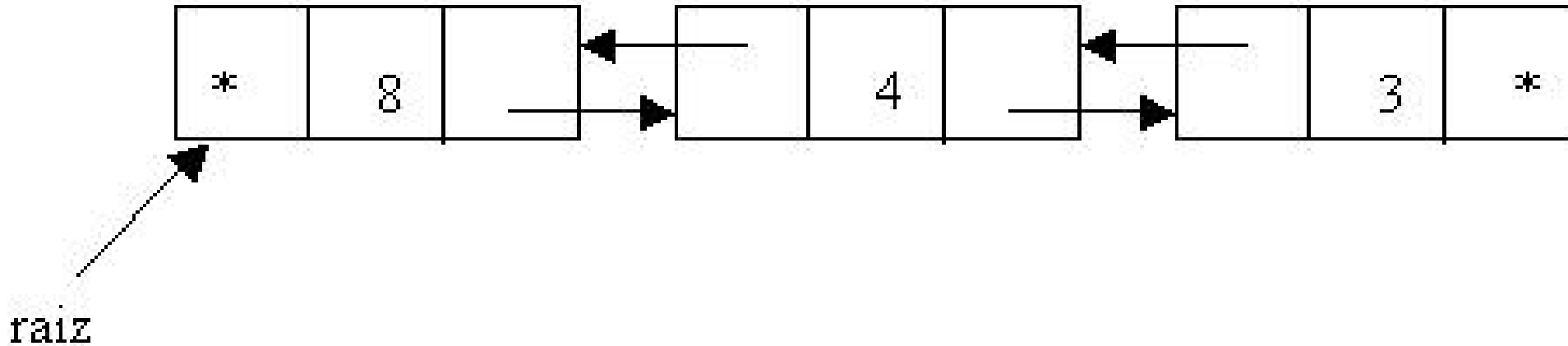
Lista ligada circular

- El ultimo nodo apunta al primer nodo.



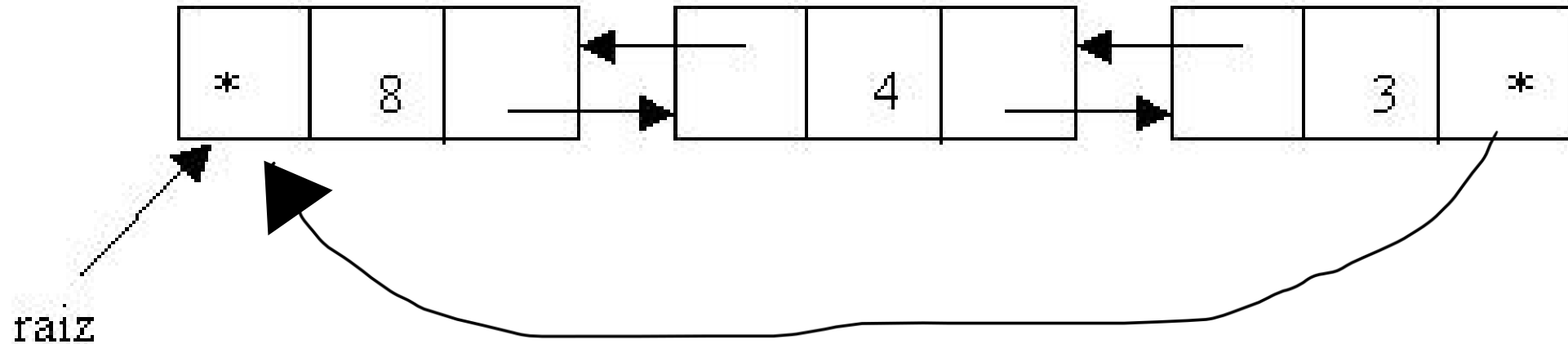
Lista doblemente ligada

- Cada nodo tiene dos punteros, uno al siguiente nodo y otro al nodo anterior



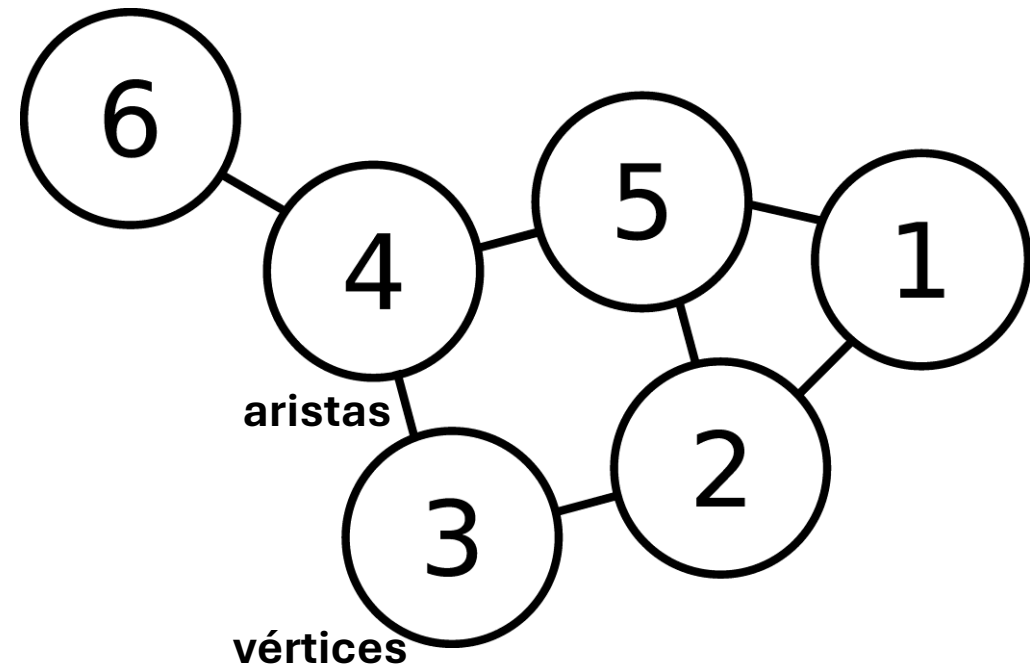
Lista doblemente ligada circular

- Cada nodo tiene dos punteros, uno al siguiente nodo y otro al nodo anterior
- El ultimo nodo apunta al primer nodo.



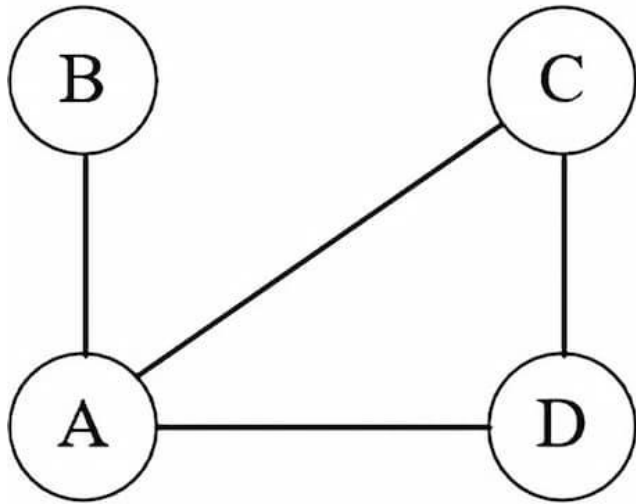
Grafos

estructura compuesta por un conjunto de **vértices** (nodos) y un conjunto de **aristas** (conexiones entre nodos).



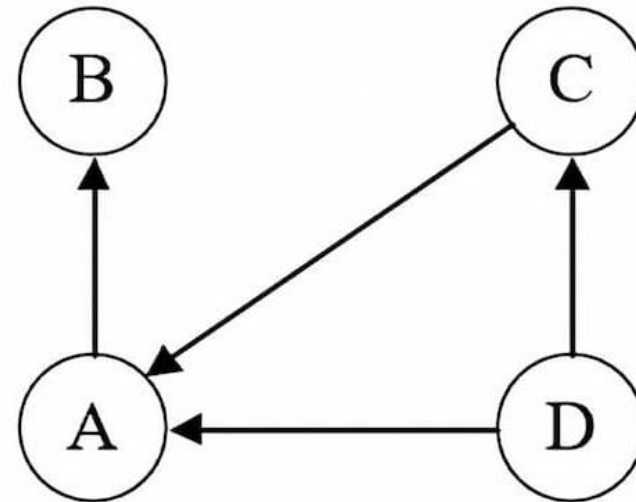
Grafos

Grafo no dirigido



Las aristas NO tienen una dirección.

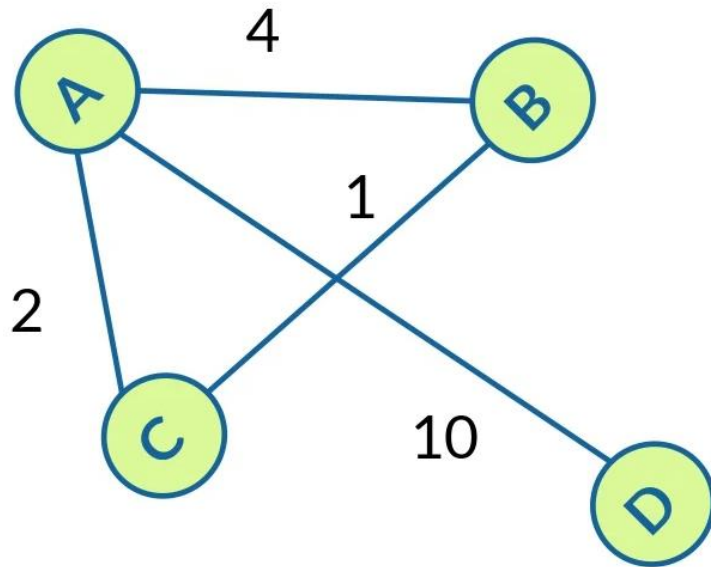
Grafo dirigido



Las aristas tienen una dirección,
representada por flechas.

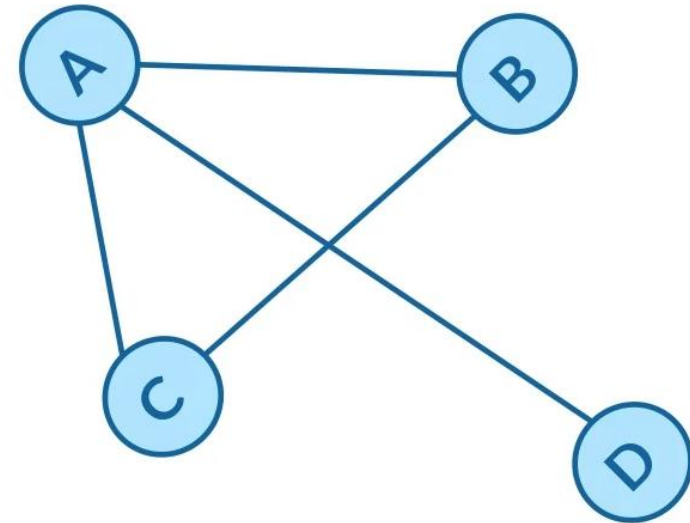
Grafos

Ponderados



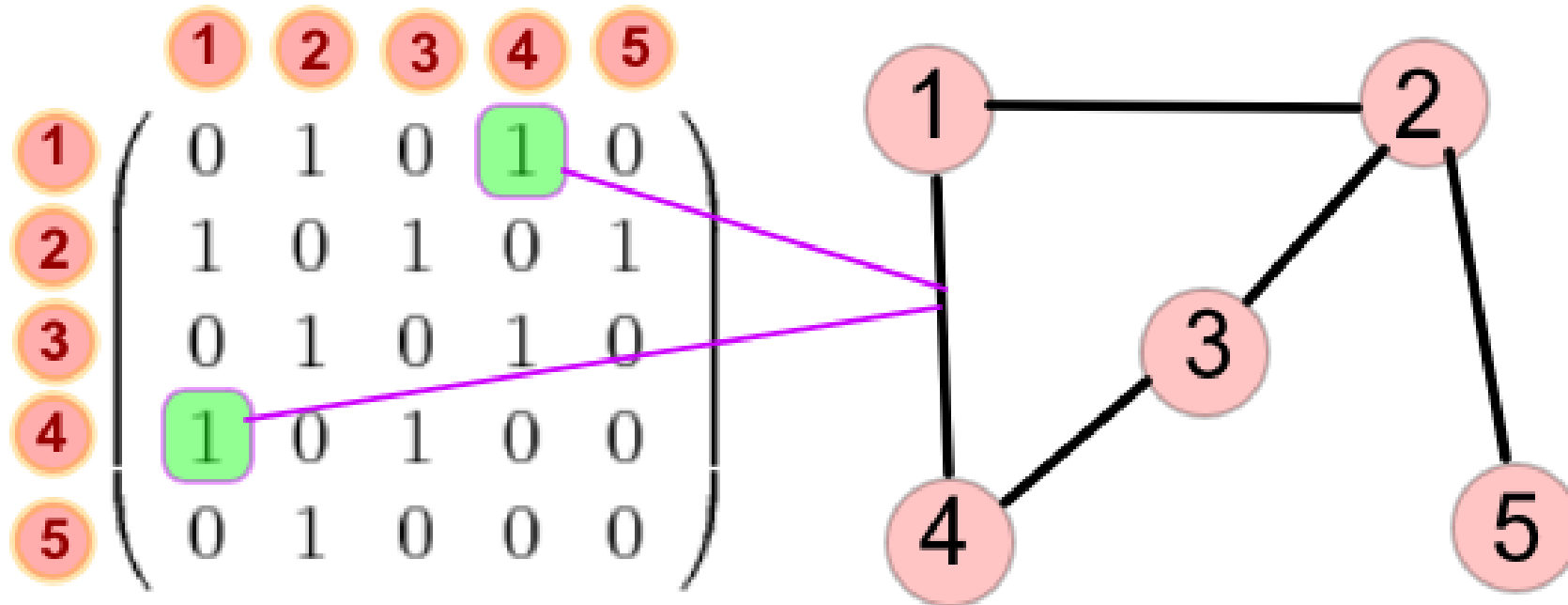
Las aristas tienen un peso o costo asociado.

No Ponderados



Las aristas NO tienen un peso asociado.

Grafos



Matriz de adyacencia: Matriz cuadrada donde cada celda $[i][j]$ indica si existe una arista entre los nodos i y j

Grafos

Operaciones y algoritmos comunes:

- **Búsqueda en profundidad (DFS) y Búsqueda en amplitud (BFS):** Algoritmos para recorrer o buscar en un grafo.
- **Dijkstra y Floyd-Warshall:** Algoritmos para encontrar caminos más cortos en grafos ponderados.
- **Kruskal y Prim:** Algoritmos para encontrar árboles de expansión mínima.

Grafos

Aplicaciones:

- Modelado de redes (redes de computadoras, redes sociales).
- Planificación de rutas (GPS, redes de transporte).
- Resolución de problemas en teoría de grafos (como el problema del viajante o el problema de coloreado de grafos).



Google Maps

Tabla de hash

estructura de datos que almacena pares clave-valor. Permite búsquedas, inserciones y eliminaciones extremadamente eficientes.

1. La función hash transforma una clave en un índice dentro del rango de la tabla.
2. Se utiliza una función hash se utiliza para calcular el índice de almacenamiento de cada clave en una tabla.

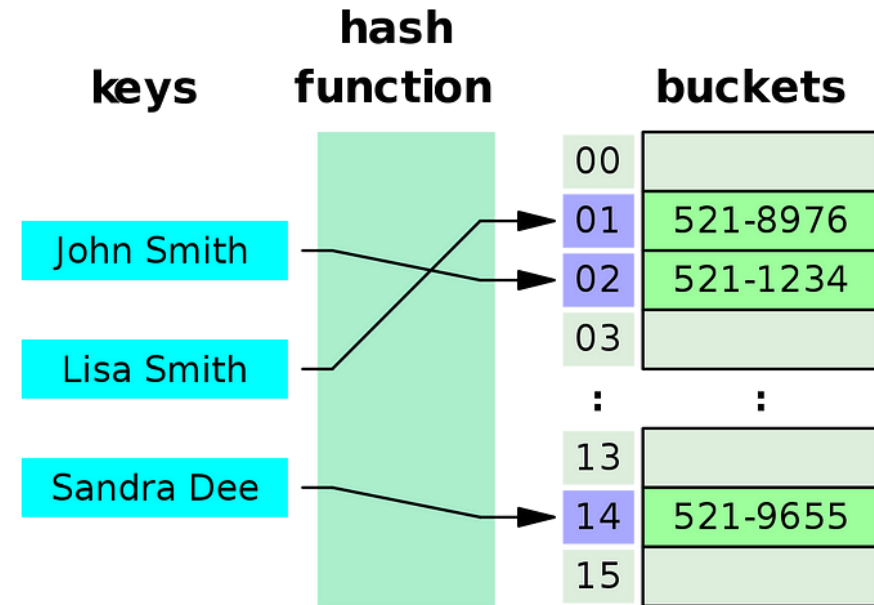


Tabla de hash

Colisiones: Ocurren cuando dos claves diferentes generan el mismo índice.

Las técnicas comunes para manejar colisiones incluyen:

- **Encadenamiento:** Se almacena una lista enlazada en cada índice donde hay más de una clave.
- **Dirección abierta:** Se busca el siguiente índice disponible siguiendo una secuencia determinada (p. ej., sondeo lineal o cuadrático).

Tabla de símbolos

estructura de datos que se utiliza en compiladores e intérpretes para almacenar información relevante sobre identificadores (variables, funciones, constantes, tipos, etc.)

Cada entrada en la tabla de símbolos generalmente contiene:

- Nombre del identificador.
- Tipo de dato (int, float, char, etc.).
- Alcance (scope) del identificador (global, local).
- Dirección de memoria o desplazamiento.
- Información adicional como el número de parámetros en funciones.

Tabla de símbolos

Operaciones principales:

- **Búsqueda:** Verificar si un identificador ya está declarado.
- **Inserción:** Añadir un nuevo identificador a la tabla cuando se declara.
- **Eliminación:** Retirar identificadores del ámbito que se abandona

Puede implementarse usando estructuras como:

- **Listas enlazadas:** Útil para manejar tablas pequeñas.
- **Tablas hash:** Eficiente en términos de tiempo de búsqueda.
- **Árboles binarios:** Útil cuando se necesita mantener el orden de los identificadores.