

DATOS MASIVOS I

UNIDAD II MODELO DE MAPEO Y REDUCCIÓN

ALGORITMOS DEL MODELO MAP – REDUCE

Sincronización de Tareas



Imagen tomada de <https://www.abc.es/>

Sincronización de Tareas

- Sincronización de tareas.
 - Las operaciones de mapeo y reducción se ejecutan de manera simultánea.
 - Los nodos de reducción reciben las llaves de manera ordenadas.
- Creación de estructuras algebraicas que contribuyan a optimizar los procesos.

Algoritmos del Modelo Map – Reduce

- El modelo de programación implementa diversos algoritmos matemáticos para dividir una tarea en pequeñas partes y para asignarlas a múltiples nodos.
 - Ordenamiento.
 - Búsqueda.
 - Índice invertido.

Ordenamiento

Tarea. Ordenar un conjunto de archivos, un valor por línea.

- Algoritmo.
 - Toma ventaja de las propiedades del reductor pares (*llave*, *valor*) las cuales son procesados en orden por *llave*. Los reductores se ordenan ellos mismos.
- Función de mapeo.
 - La llave será el nombre de archivo y número de línea, el valor será el contenido de línea.
 - Regresa el valor como llave $(k, v) \Rightarrow (v,)$
- Función de reducción.
 - Función de identidad.

Ordenamiento

Aprovecha el ordenamiento de llaves por sistema de manejo de tareas, se define una función de partición tal que:

$$k_1 < k_2 \implies \text{hash}(k_1) < \text{hash}(k_2)$$

- Es usado por prueba de velocidad de Hadoop.
- Es un carrera de resistencia *entradas – salidas*.

Búsqueda

Tarea. Encontrar documentos que contienen un patrón dado.

- Función de mapeo.
 - La llave será el nombre de archivo y número de línea, el valor será el contenido de línea.
 - Regresa nombre de archivo como llave si se encuentre el patrón en el contenido.

- Función de reducción.
 - Identidad.

Búsqueda

- Una vez que se identificó al documento con el patrón, es necesario marcar ese documento (una sola vez).
- Se usa la función *Combiner* para convertir pares redundantes en un solo archivo (*filename,*).
- Reduce la entradas / salidas de la red.

Combiner Function

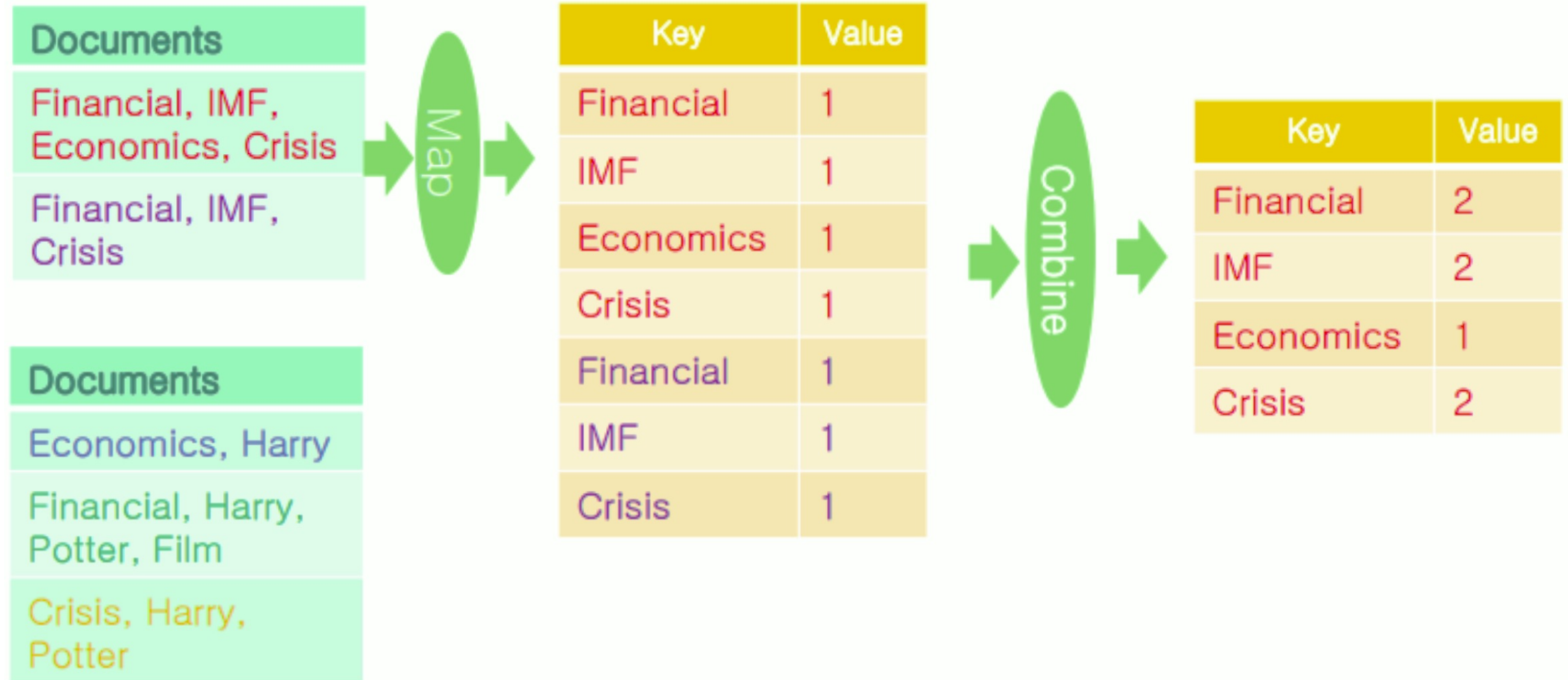


Imagen tomada de Kyuseok Shim, 2013.

Combiner Function



Imagen tomada de Kyuseok Shim, 2013.

Combiner Function

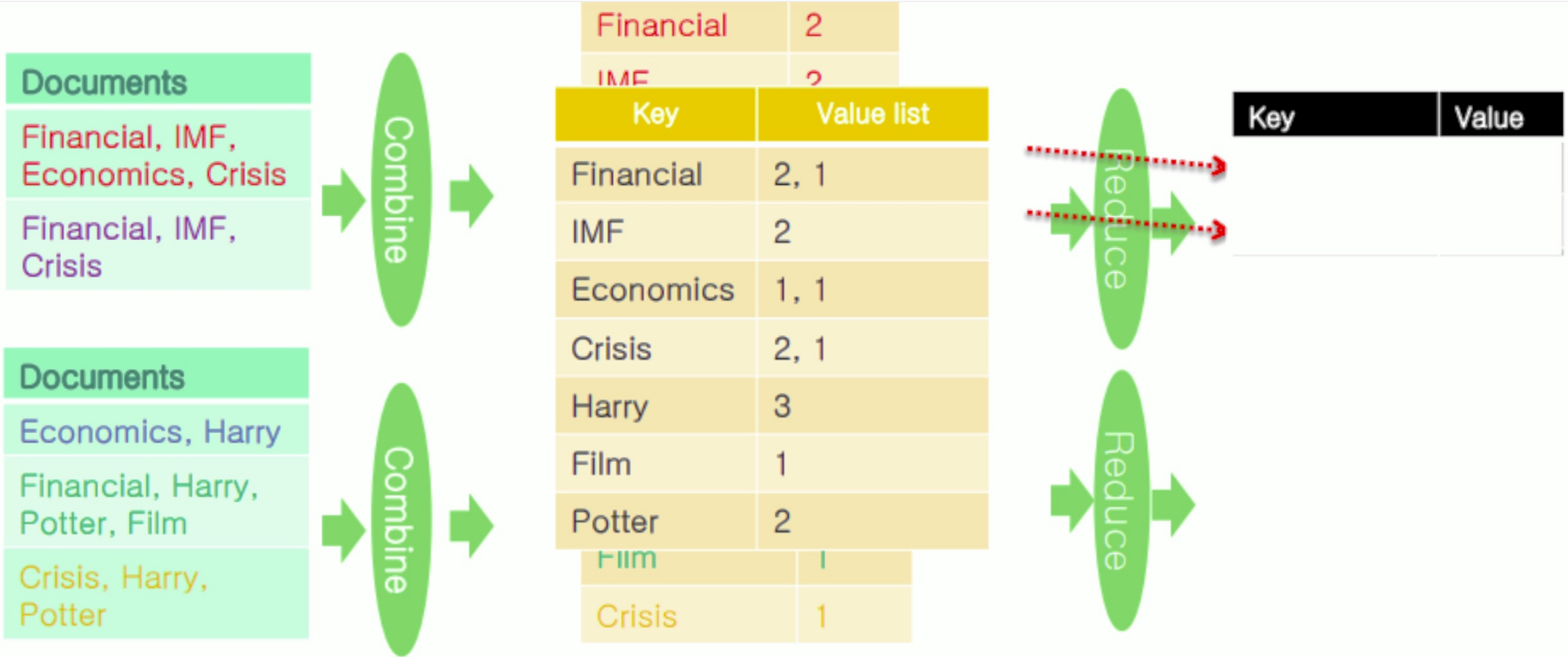


Imagen tomada de Kyuseok Shim, 2013.

Índice Invertido

- Este algoritmo es el más utilizado en los sistemas de recuperación de información (por ejemplo, motores de búsqueda).

¿Cómo trabaja este algoritmo?

- Para cada término ***t***, guardamos todos los documentos que contienen ***t***.
- Identificamos cada documento por un ***docID***, el cuál es un número incremental.

Índice Invertido

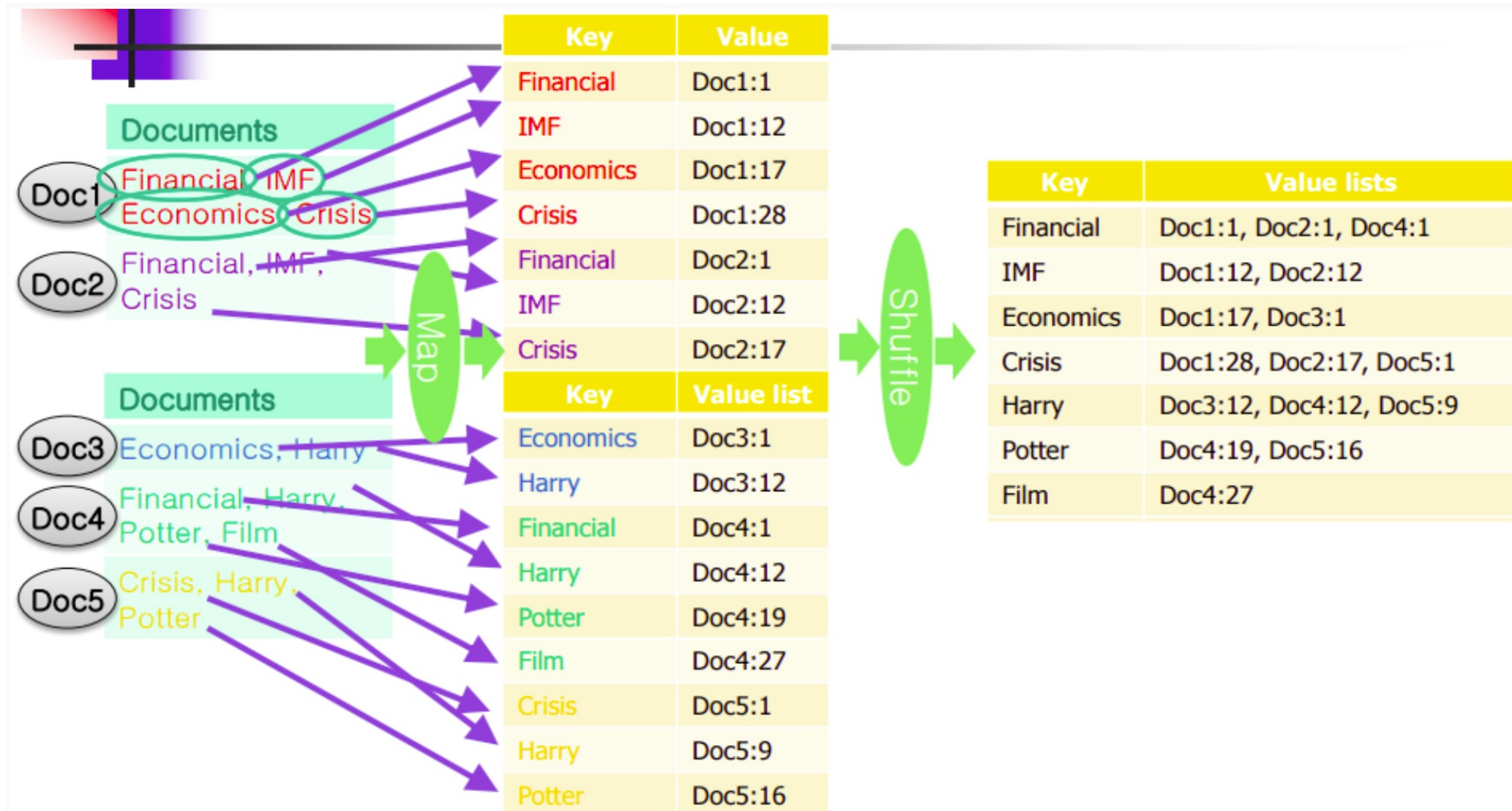


Imagen tomada de Kyuseok Shim, 2013.

Grafos, ¿Qué son?

$$G = (V, E)$$

- V representa un conjunto de vértices (nodos).
- E representa un conjunto de aristas (enlaces).
- Tanto vértices como aristas pueden contener información adicional.

Tipos de grafos.

- Dirigidos o no dirigidos.
- Con y sin ciclos.

Los G están en todas partes:

- Redes sociales.
- Estructura de hipervínculos en la web.
- Estructuras físicas de las computadoras.

Algunos Problemas con los Grafos

- Encontrar la ruta más corta.
 - Tráfico en internet.
- Búsqueda de árboles mínimos.
 - Tendido de fibra óptica.
- Encontrar el flujo máximo.
 - Programación de aerolíneas.
- Identificar comunicaciones 'especiales'.
 - Terrorismo / conspiraciones.
 - Enfermedades.

Grafos y Map – Reduce

El procesamiento con grafos:

- ✓ Ejecutar cálculos en cada uno de los nodos: basados en características de los nodos y de las aristas.
- ✓ Propagar los cálculos 'atravesando' el grafo.

Preguntas claves:

- ¿Cómo representar los datos de un grafo en Map – Reduce?
- ¿Cómo recorrer el grafo usando Map – Reduce?

Grafos y Map – Reduce

Los grafos comúnmente son representados como una matriz de adyacencias o una lista de adyacencias.

	1	2	3	4
1	0	1	0	1
2	1	0	1	1
3	1	0	0	0
4	1	0	1	0

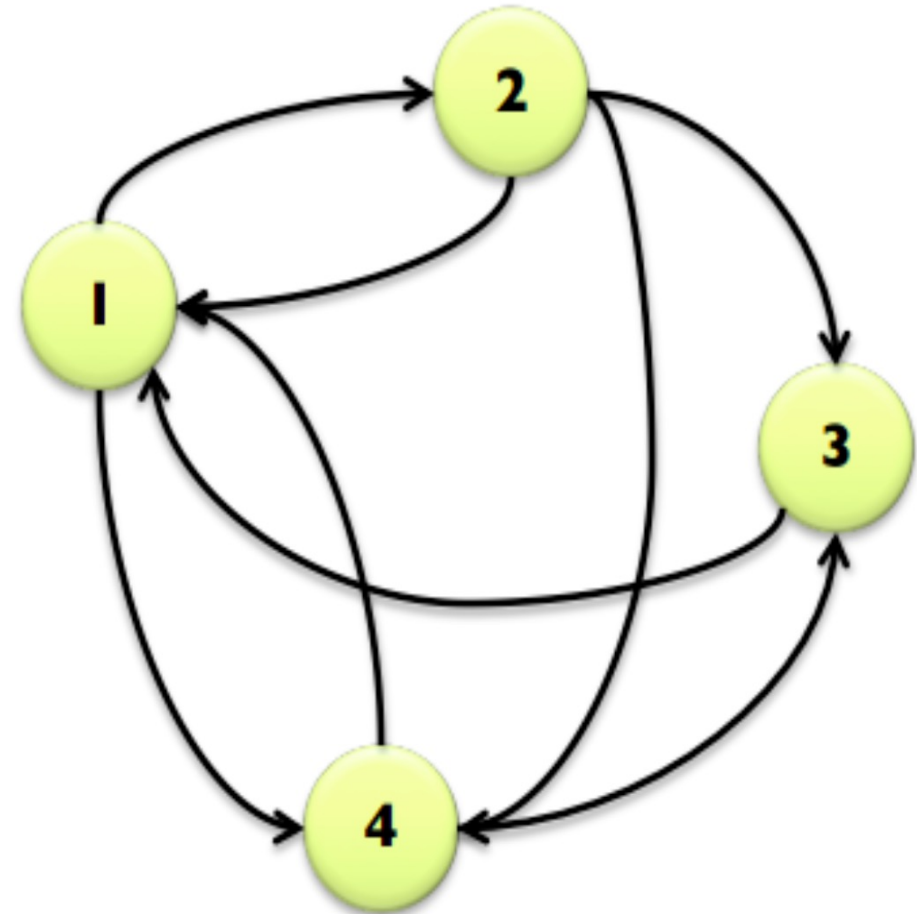


Imagen tomada de Jimmy Lin, 2013.

Matrices de Adyacencia: Pros and Cons

- Ventajas.
 - ✓ Manipulación accesible de los datos.
 - ✓ Iteración sobre filas y columnas, correspondientes con los enlaces salientes y entrantes.
- Desventajas.
 - Muchos ceros.
 - Ocupan una gran cantidad de espacio.

Listas de Adyacencia

	1	2	3	4
1	0	1	0	1
2	1	0	1	1
3	1	0	0	0
4	1	0	1	0



1: 2, 4
2: 1, 3, 4
3: 1
4: 1, 3

Imagen tomada de Jimmy Lin, 2013.

Encontrar el Camino Más Corto (Shortest Path: BFS)

```
1: class MAPPER
2:   method MAP(nid  $n$ , node  $N$ )
3:      $d \leftarrow N.DISTANCE$ 
4:     EMIT(nid  $n$ ,  $N$ )                                ▷ Pass along graph structure
5:     for all nodeid  $m \in N.ADJACENCYLIST$  do
6:       EMIT(nid  $m$ ,  $d + 1$ )                            ▷ Emit distances to reachable nodes

1: class REDUCER
2:   method REDUCE(nid  $m$ , [ $d_1, d_2, \dots$ ])
3:      $d_{min} \leftarrow \infty$ 
4:      $M \leftarrow \emptyset$ 
5:     for all  $d \in \text{counts } [d_1, d_2, \dots]$  do
6:       if ISNODE( $d$ ) then
7:          $M \leftarrow d$                                 ▷ Recover graph structure
8:         else if  $d < d_{min}$  then                        ▷ Look for shorter distance
9:            $d_{min} \leftarrow d$ 
10:     $M.DISTANCE \leftarrow d_{min}$                         ▷ Update shortest distance
11:    EMIT(nid  $m$ , node  $M$ )
```

Las tareas de mapeo y reducción:

- Son hilos independientes en cada nodo.

Combine Functions (funciones de combinar):

- Reduce el tamaño de las funciones de mapeo.
- Ejecuta mini-funciones de reducción en cada nodo.
- Disminuye el costo para el ordenamiento.

Cada tarea de mapeo y reducción puede opcionalmente usar dos funciones: *init()* y *close()*:

- *init()* llamado al inicio de cada tarea.
- *close()* llamado al final de cada tarea.