

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación



IIC2115 - Programación como Herramienta para la Ingeniería

Estructuras de datos

Profesor: Hans Löbel

Tipos de dato

- Tipos numéricos

int

enteros

float

racionales

- Tipos textuales

str

texto

- Tipos lógicos (booleanos)

bool

lógico

Operadores

- Aritméticos (para ints y floats principalmente, pueden definirse para otros)



Suma



Resta o
Inverso
aditivo



Producto



División



División
Entera



Resto



Potencia

- Comparación



Menor



Menor
o igual



Mayor



Mayor
o igual



Igual



Distinto

Operadores

- Lógicos (para bools)

not

Negación

and

Y

or

O

A	B	A AND B	A OR B	NOT A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

- Texto (para strs)

+

Concatenación

*

Repetición

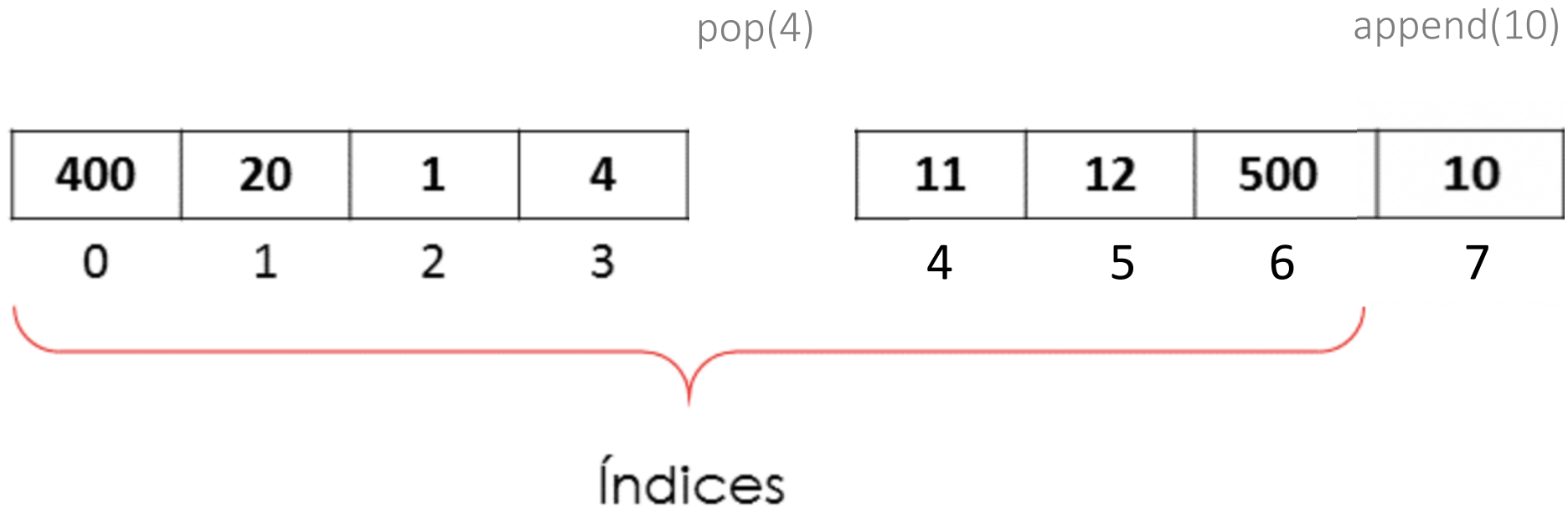
Estructuras de datos

Son tipos de dato especializados, diseñados para agrupar, almacenar o acceder a la información de manera más eficiente que un tipo de dato básico (como int, float, etc). Algunos ejemplos son los siguientes:

- Clases
- Listas
- Tuplas
- Diccionarios y sets
- Stacks y colas
- Árboles

Listas


- Las listas son estructuras que guardan datos de forma **ordenada**.
- Son mutables (modificables).



Tuplas

- Similares a las listas, permiten manejar datos de forma ordenada.
- Al igual que las listas, se accede a los datos mediante índices basados en el orden que fueron ingresados.
- A diferencia de las listas, son **inmutables**.

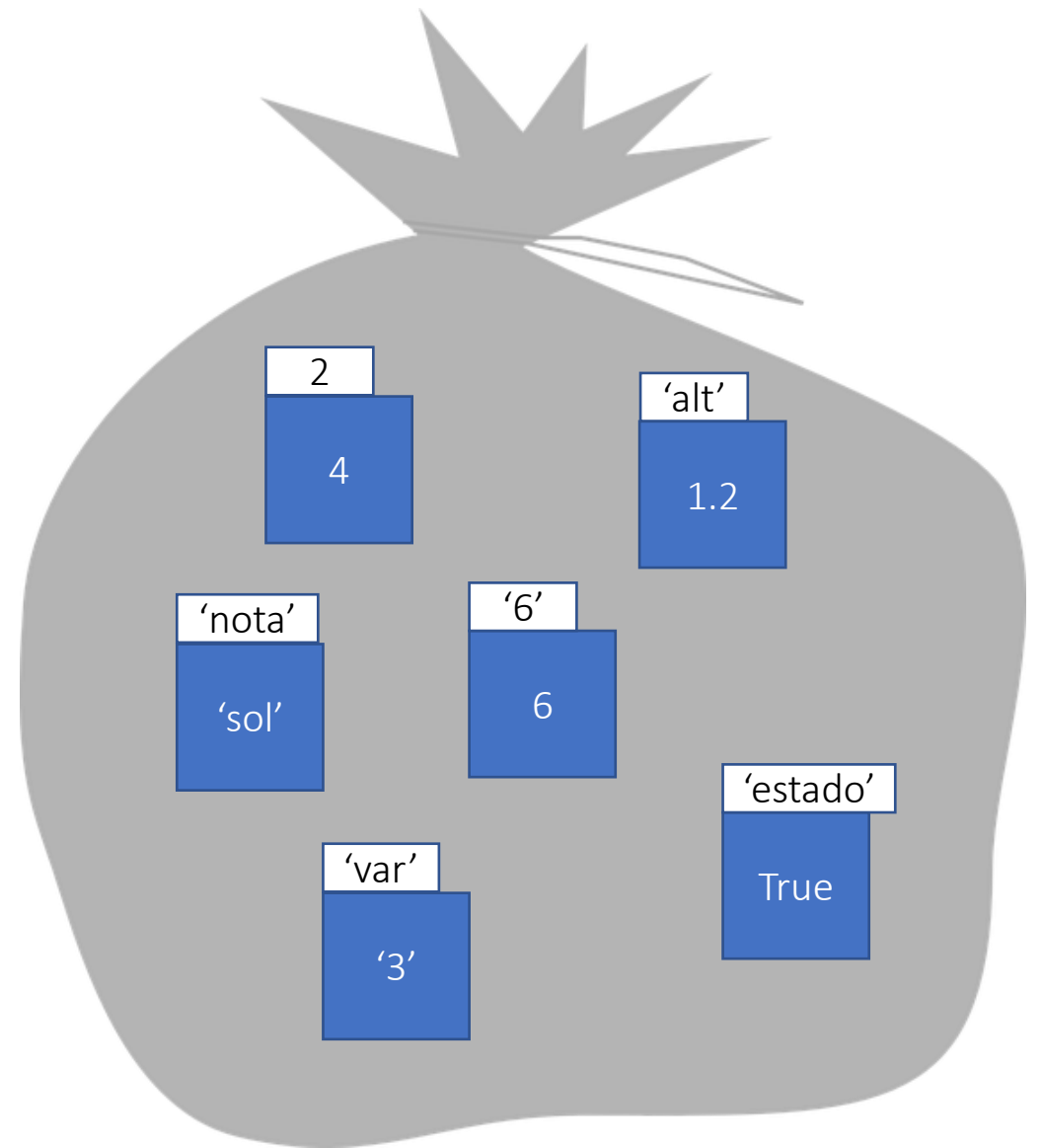
400	20	1	4	10	11	12	500
0	1	2	3	4	5	6	7



Índices

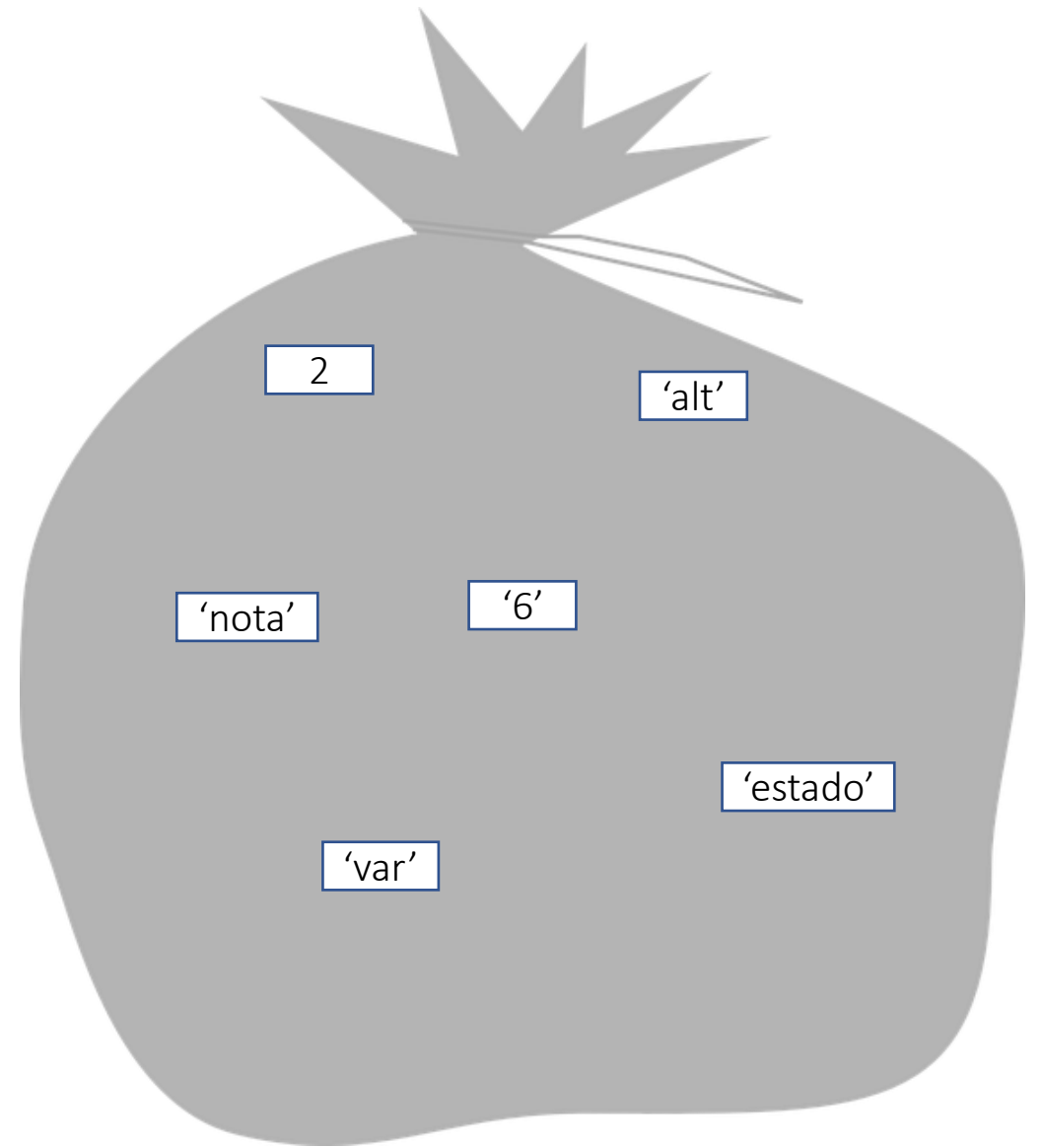
Diccionarios

- Permiten almacenar datos basados en una asociación de pares de elementos, a través de una relación **llave-valor**.
- Acceso a valores a través de la llave es instantáneo, no se necesita realizar una búsqueda (análogo a un índice).
- Se prefiere a una lista cuando el caso de uso más común no implica revisar todos los elementos, sino solo algunos fácilmente encontrables a través de la llave.



Sets

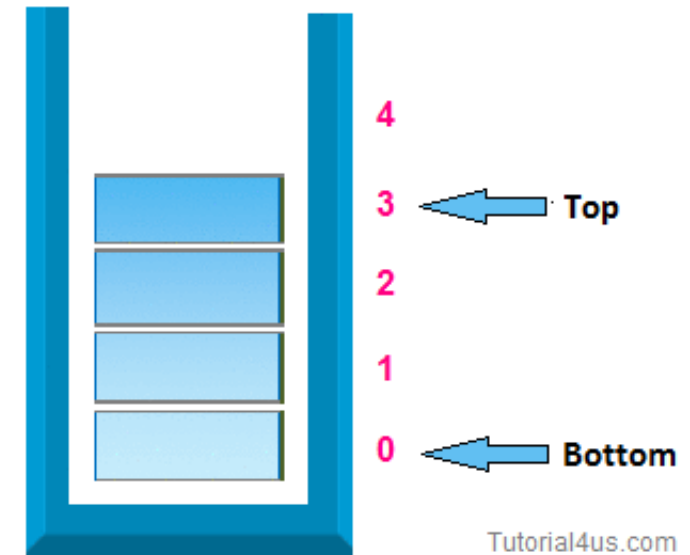
- Son como los diccionarios, pero solo consideran la llave (no hay valor asociado).
- Ideales para verificar la existencia de algo.
- Al igual que los diccionarios, no hay llaves repetidas.



Stacks y colas

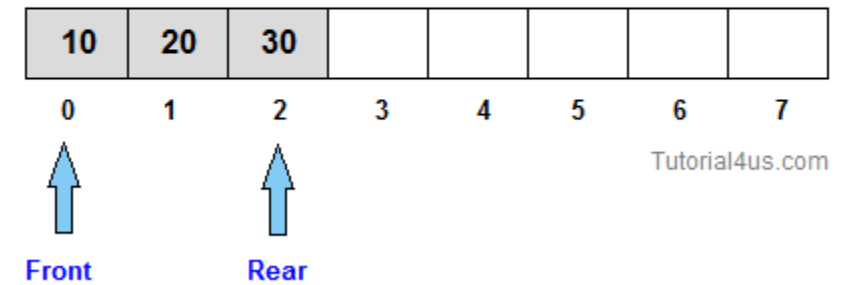
- Ordenados linealmente como las listas, pero con reglas estrictas para la extracción de elementos
- Stacks: LIFO (implementados con list o dequeue)

4	6	12	21	...	1	1.2
[0]	[1]	[2]	[3]		[-2]	[-1]

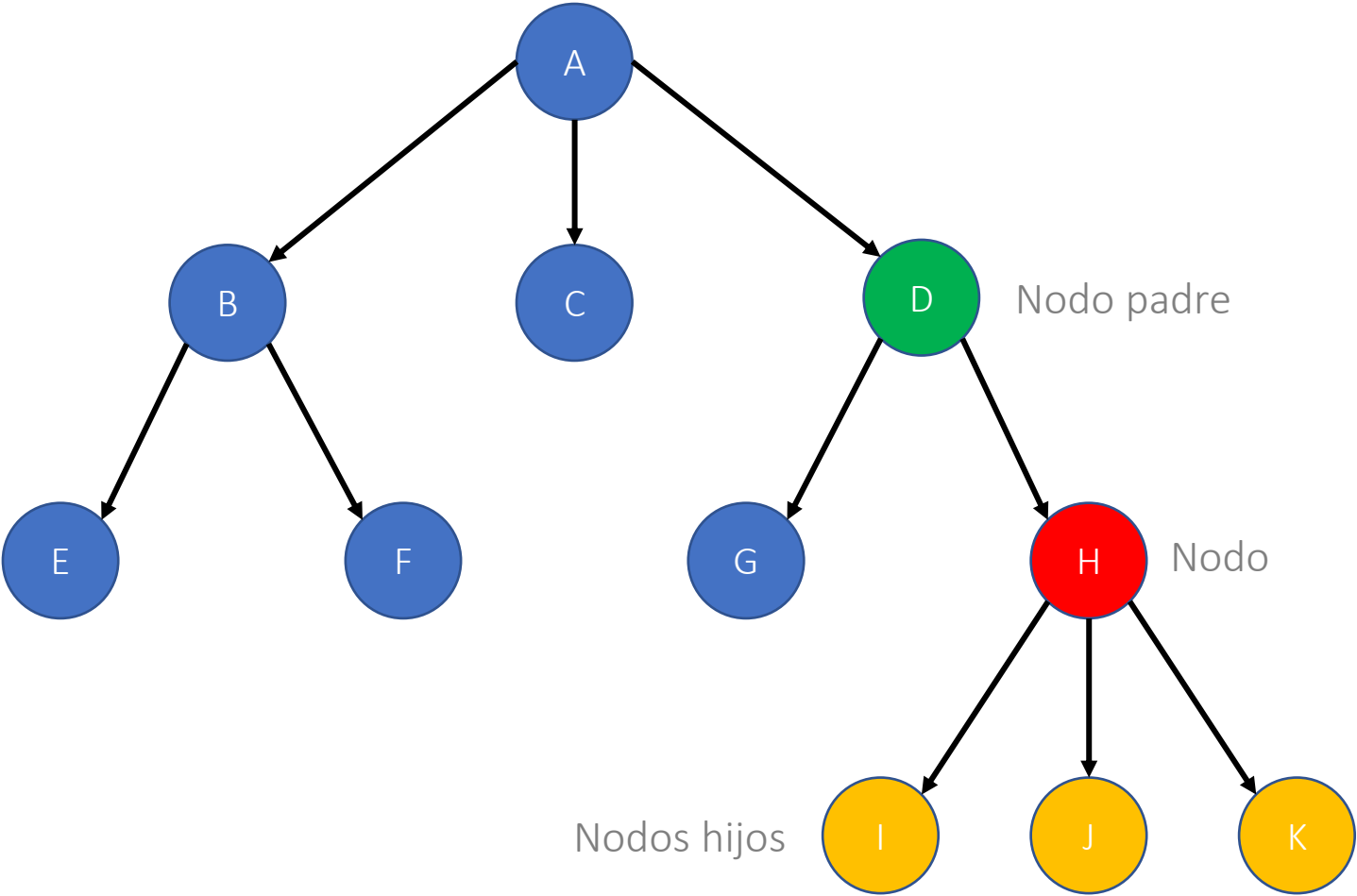


Stacks y colas

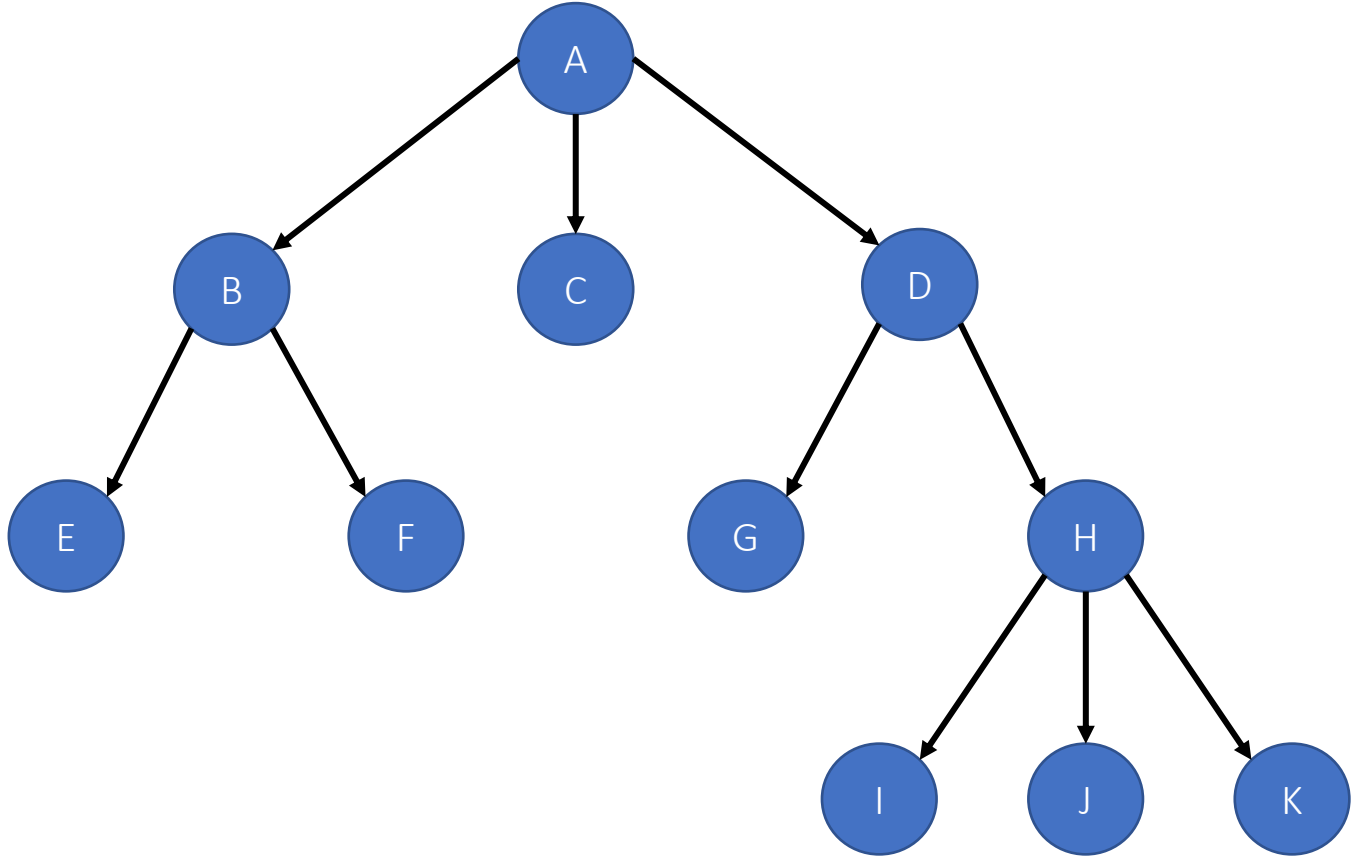
- Ordenados linealmente como las listas, pero con reglas estrictas para la extracción de elementos
- Colas: FIFO (implementadas con dequeue)



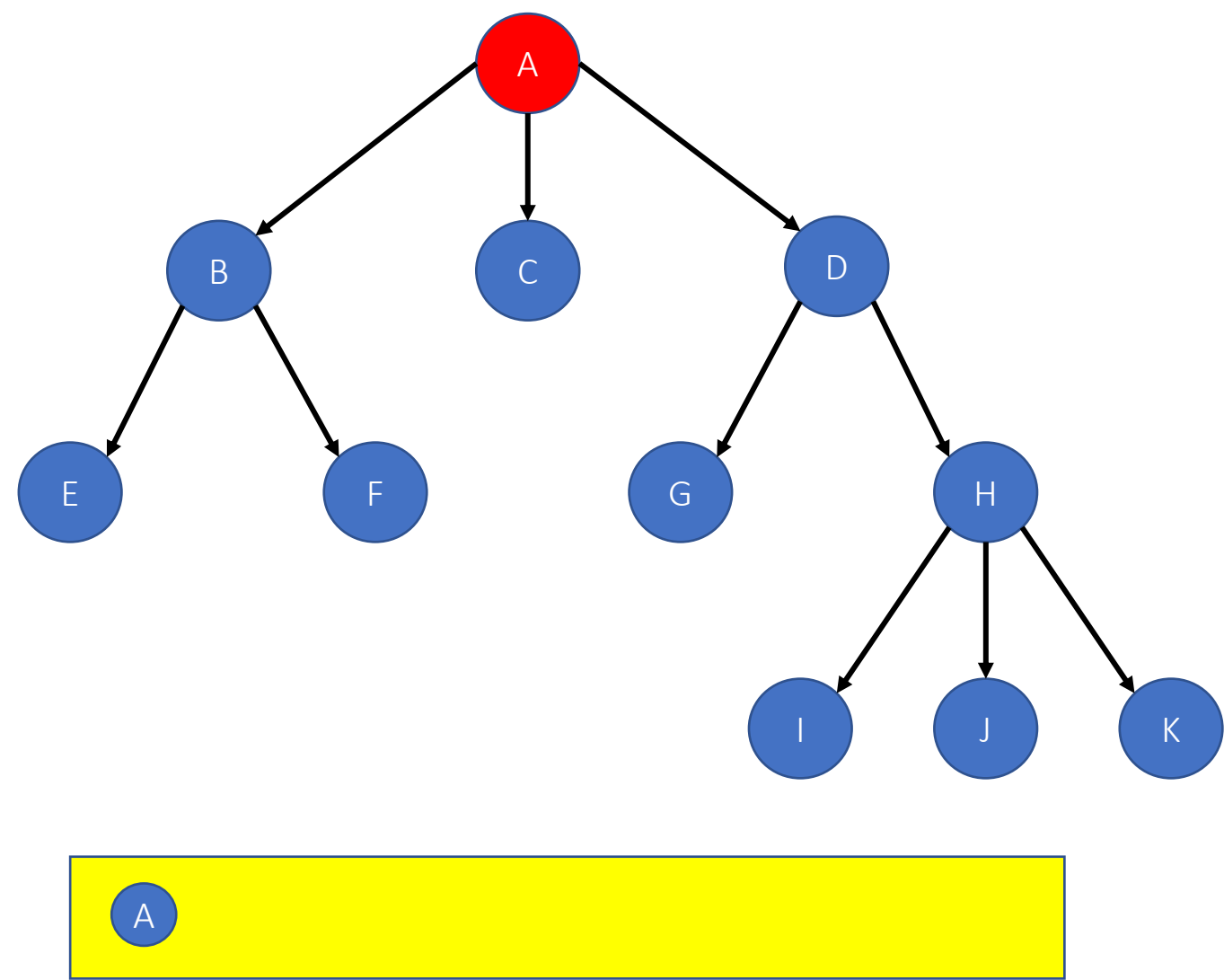
Árboles



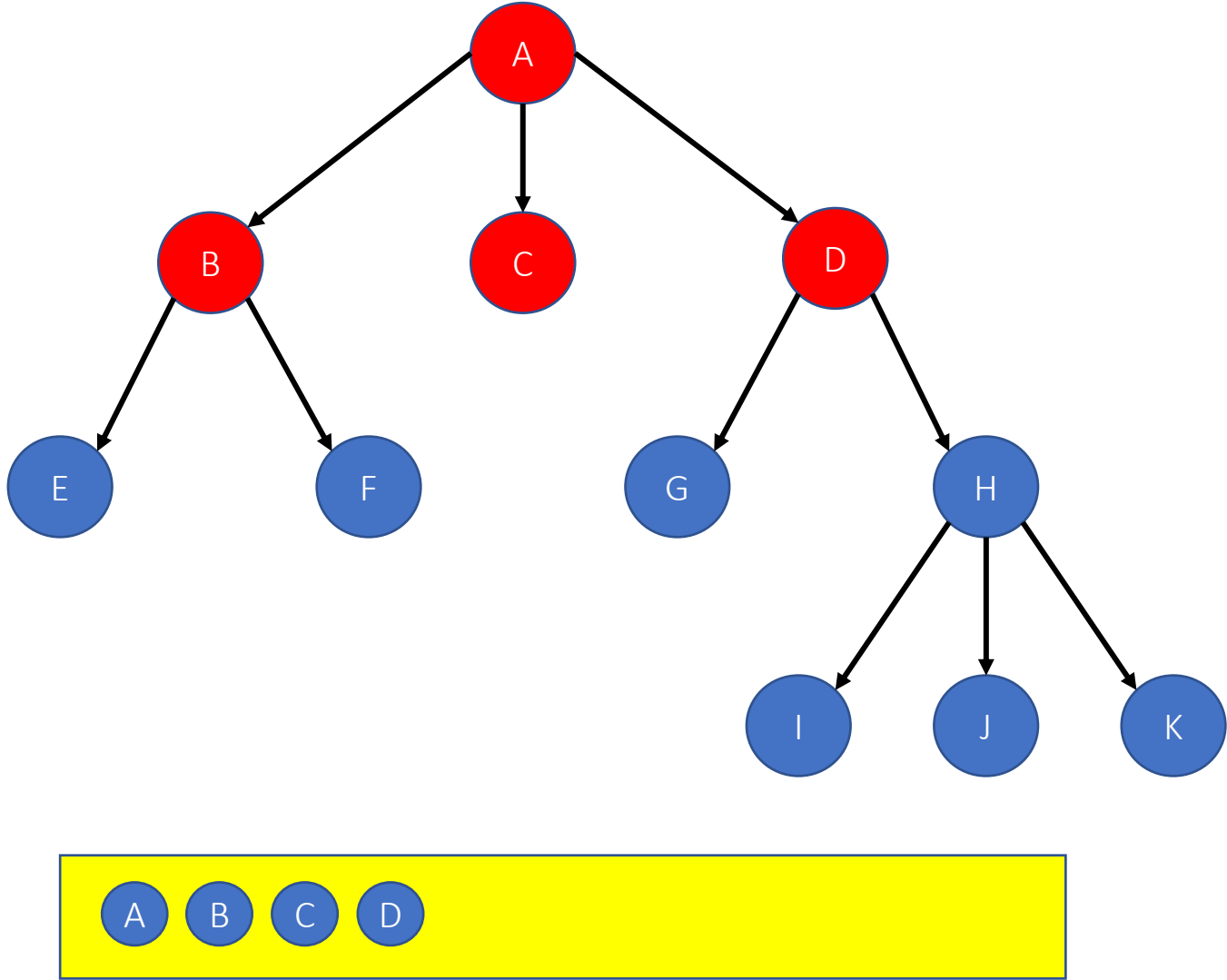
Recorriendo árboles – BFS (búsqueda en amplitud)



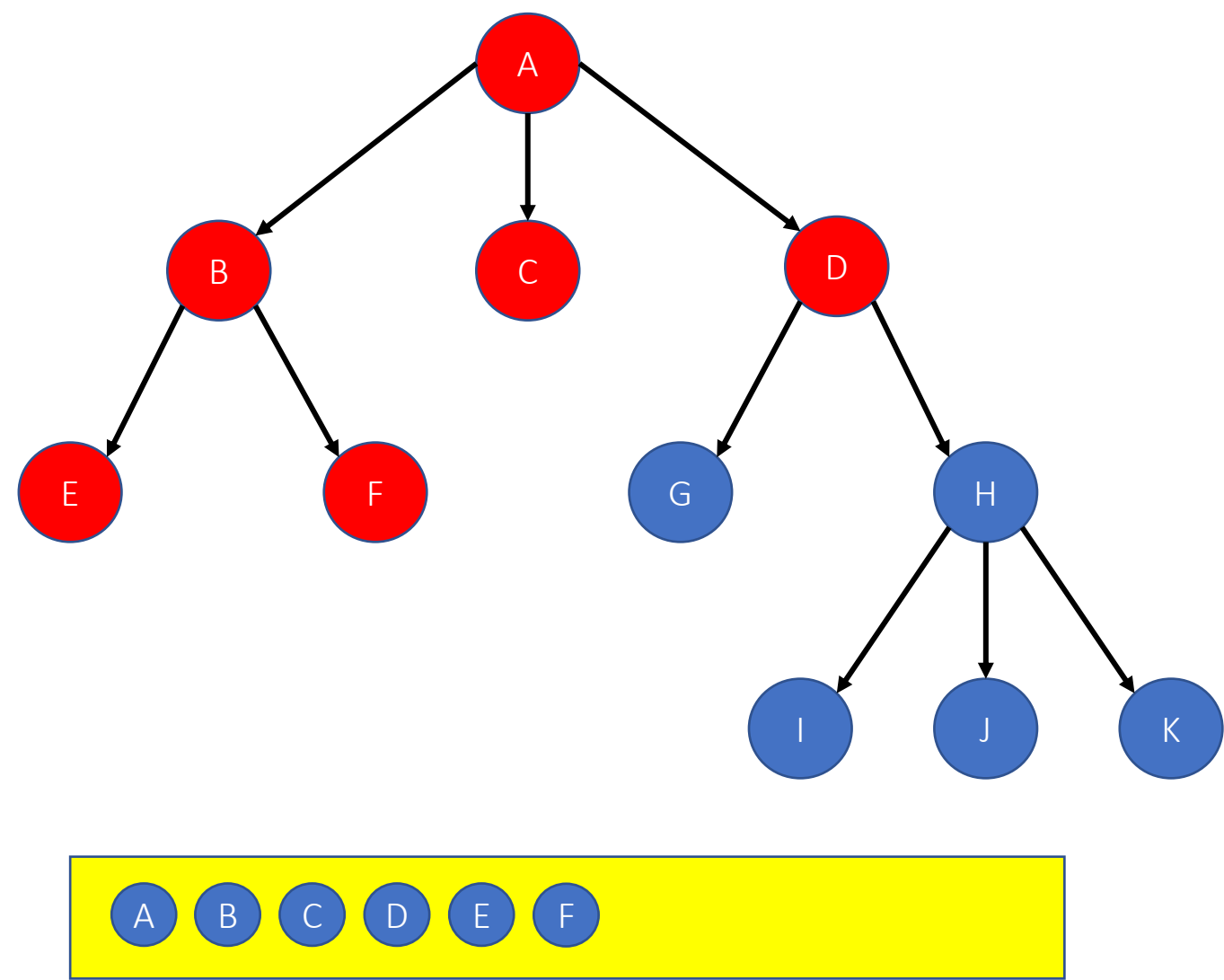
Recorriendo árboles – BFS (búsqueda en amplitud)



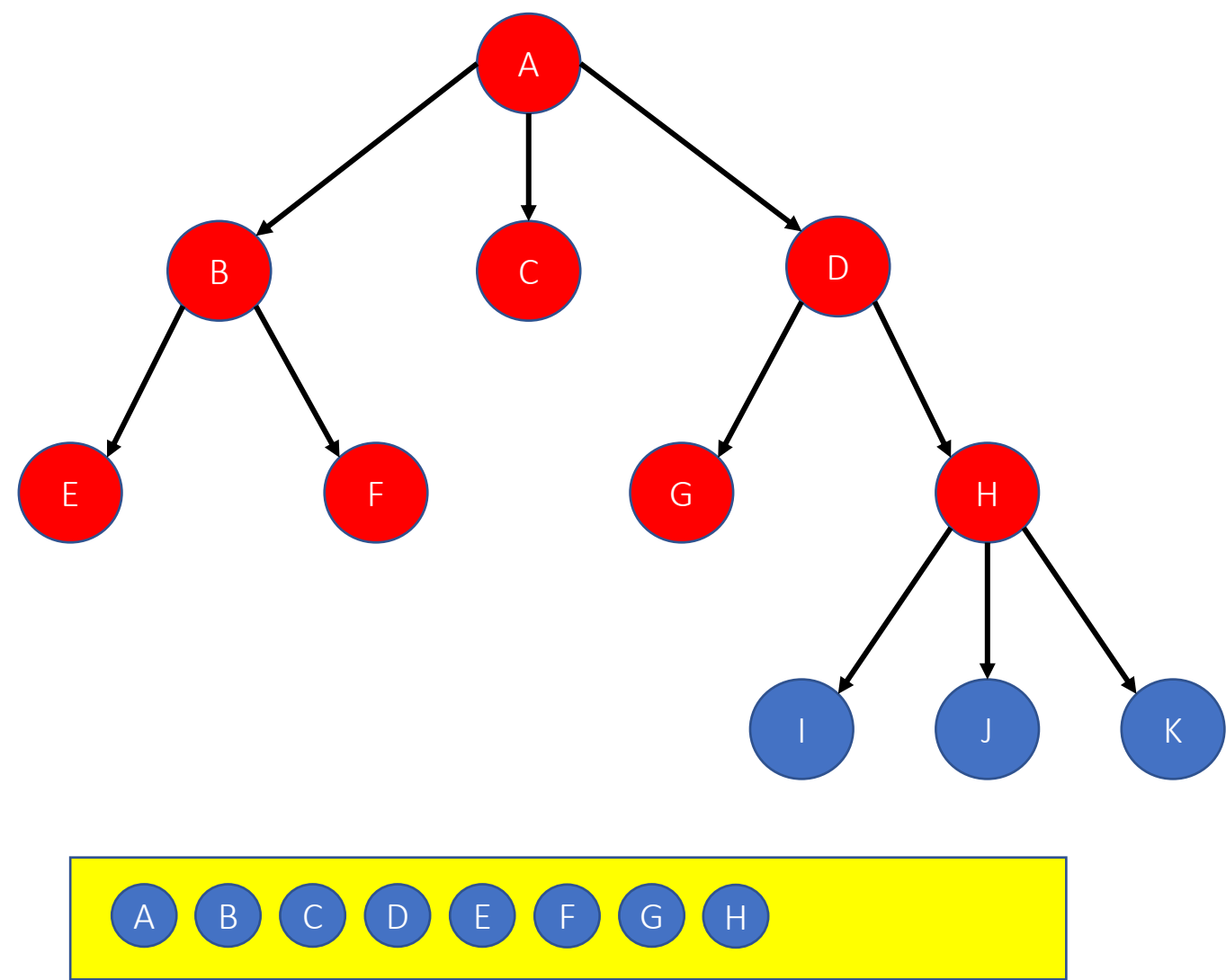
Recorriendo árboles – BFS (búsqueda en amplitud)



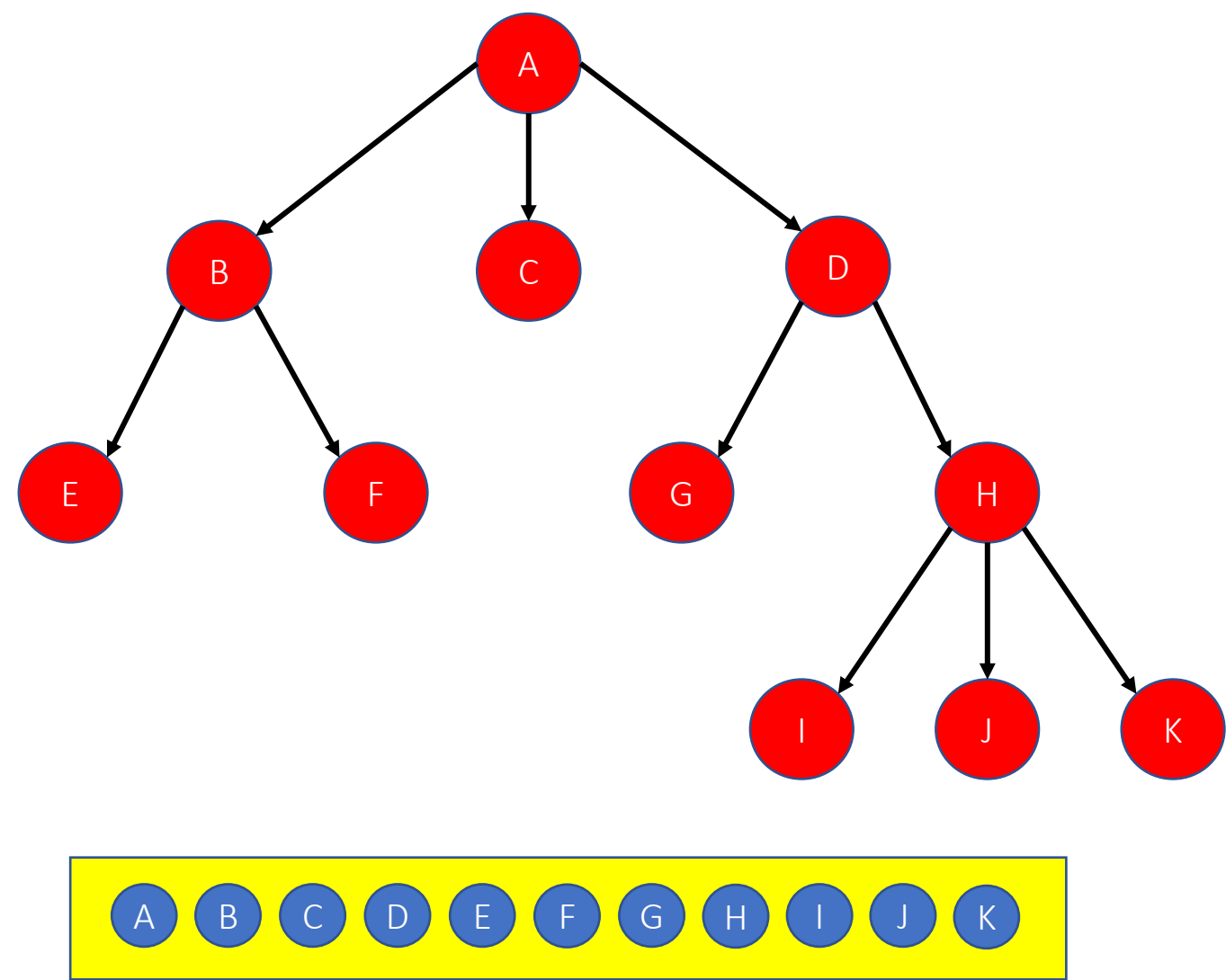
Recorriendo árboles – BFS (búsqueda en amplitud)



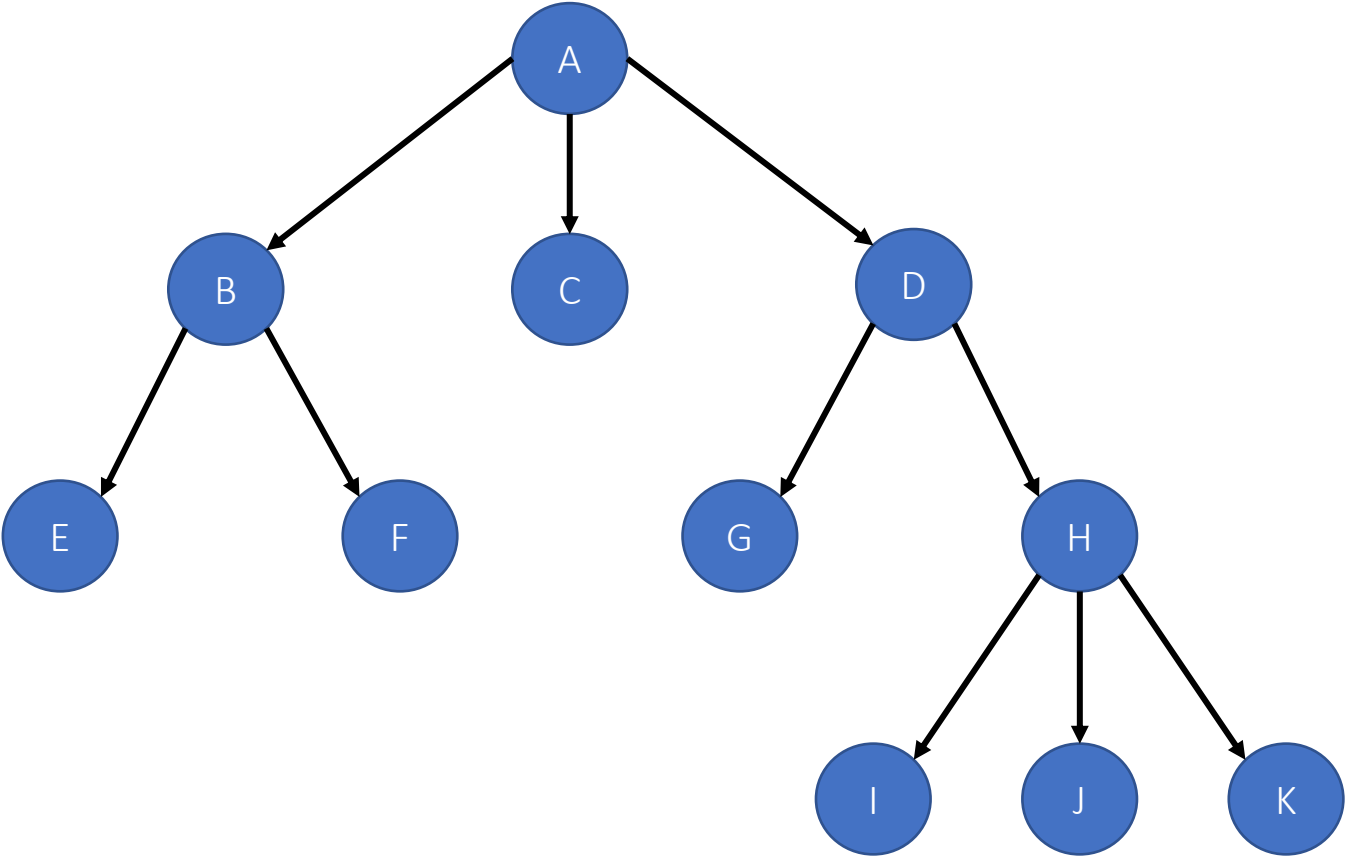
Recorriendo árboles – BFS (búsqueda en amplitud)



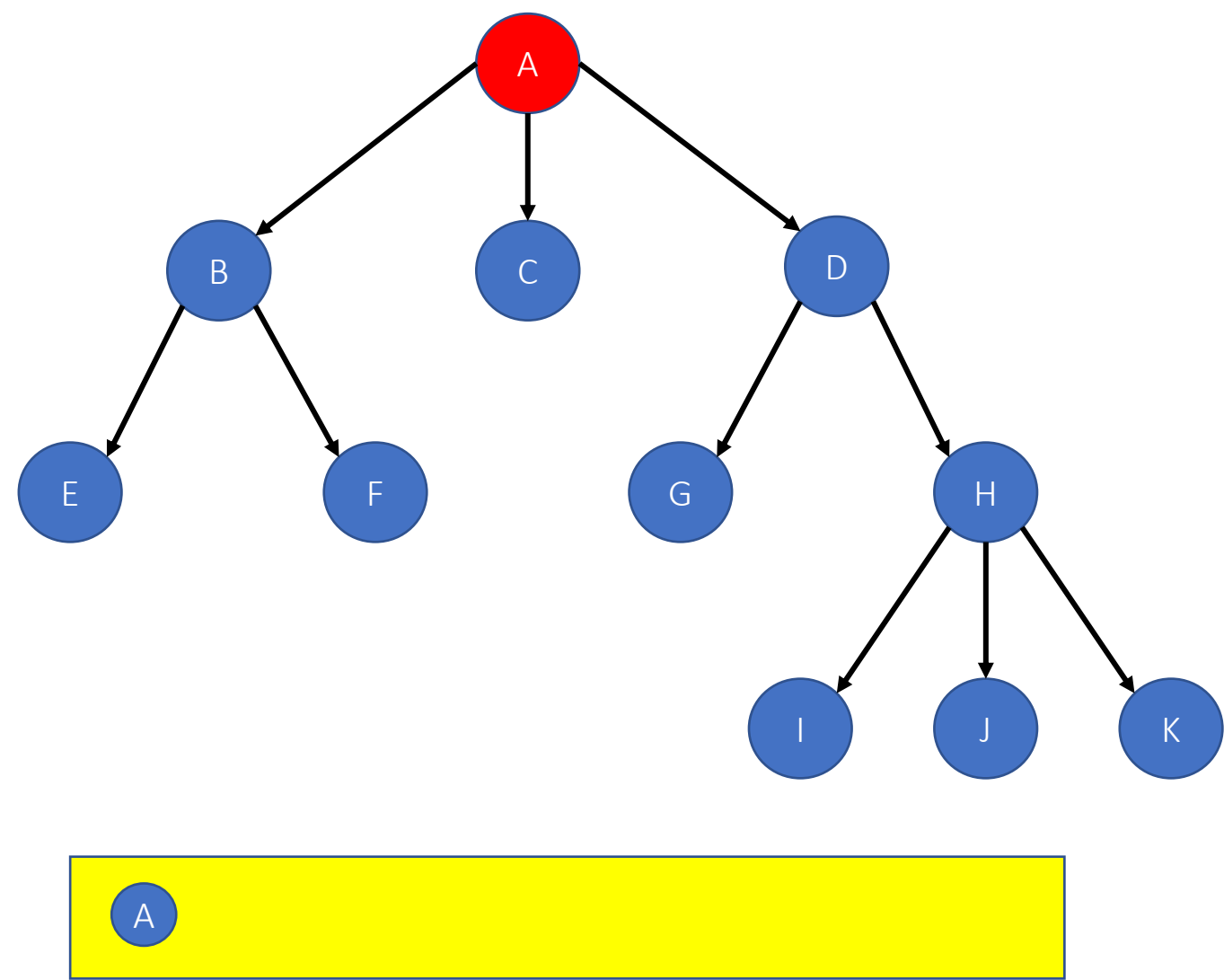
Recorriendo árboles – BFS (búsqueda en amplitud)



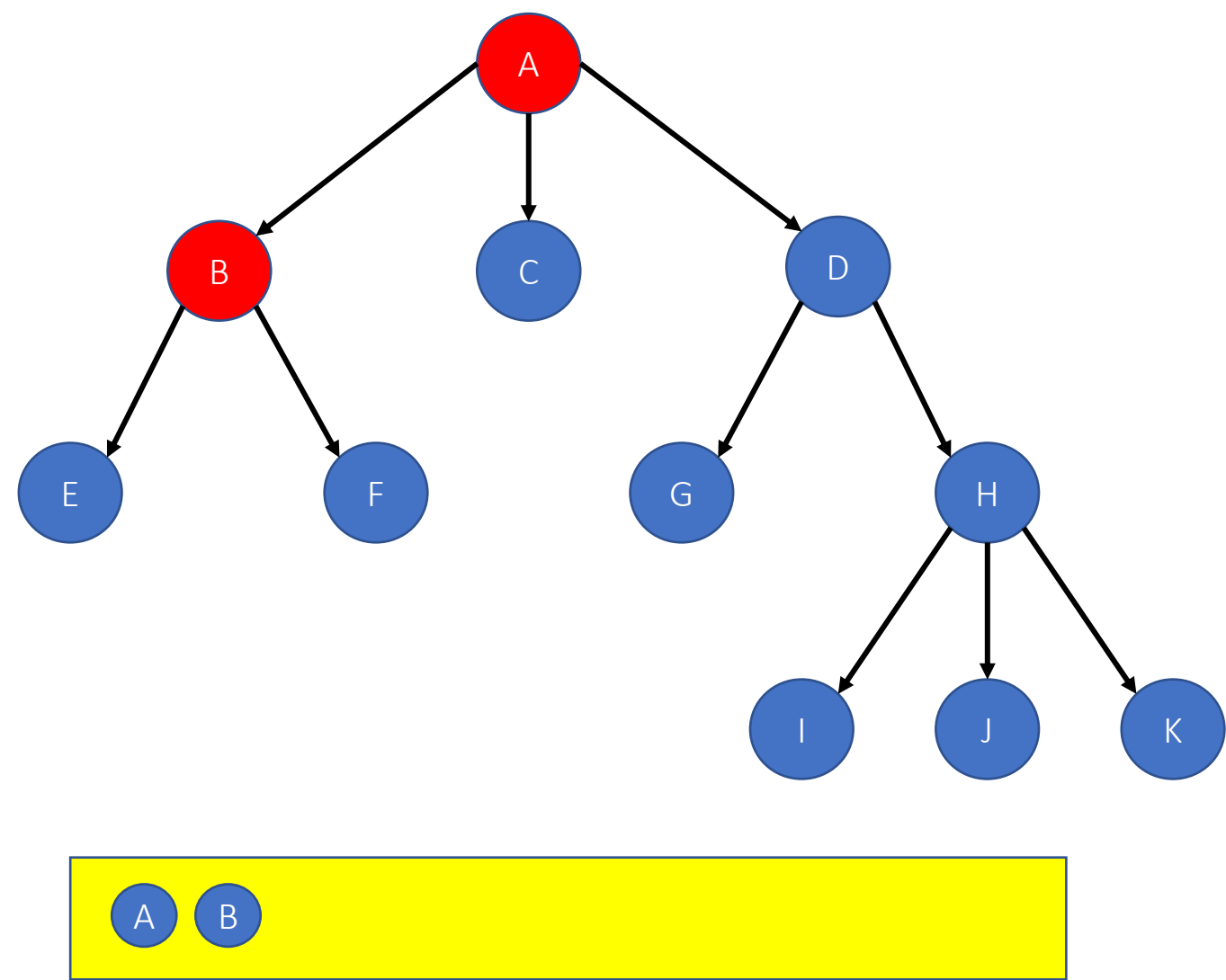
Recorriendo árboles – DFS (búsqueda en profundidad)



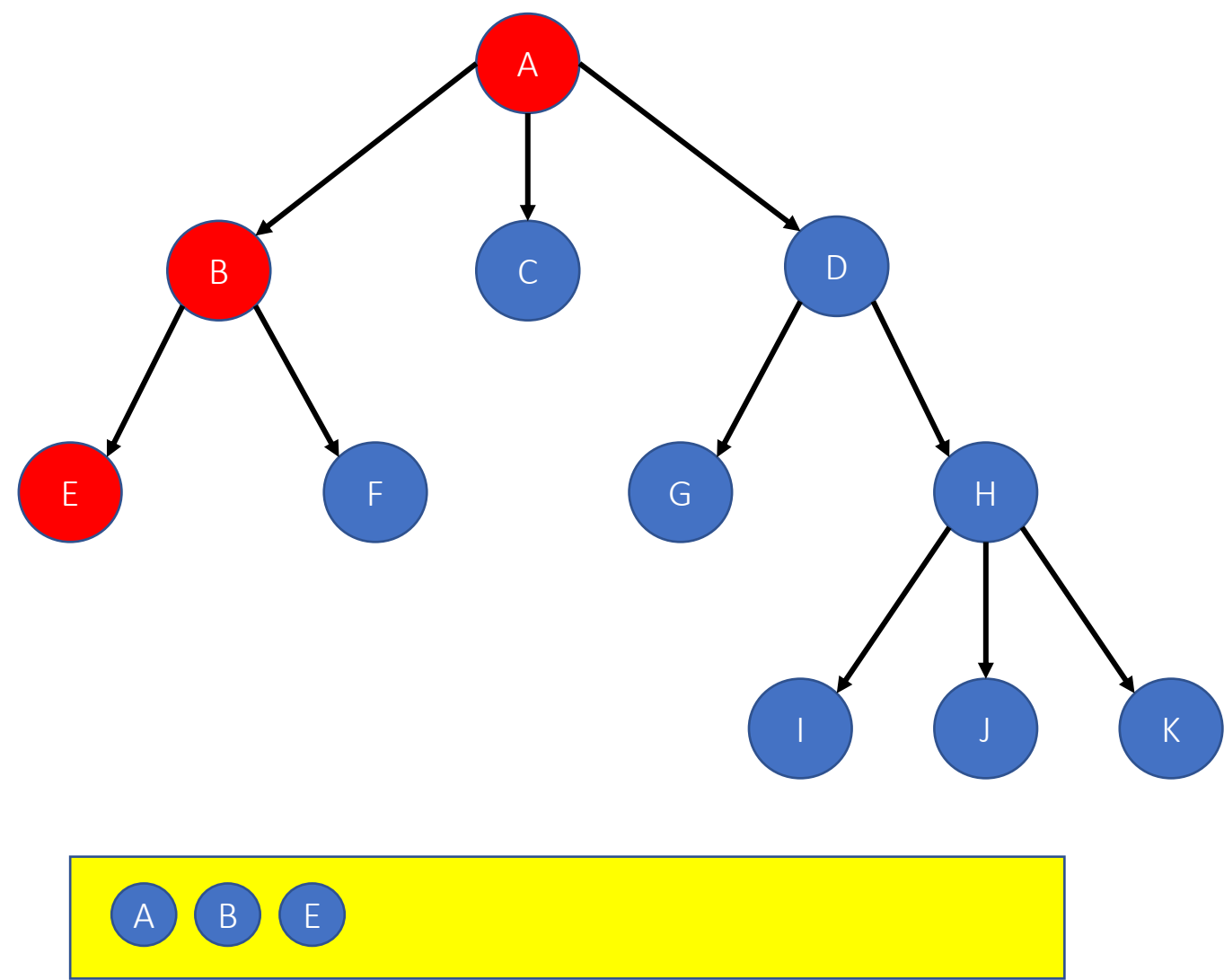
Recorriendo árboles – DFS (búsqueda en profundidad)



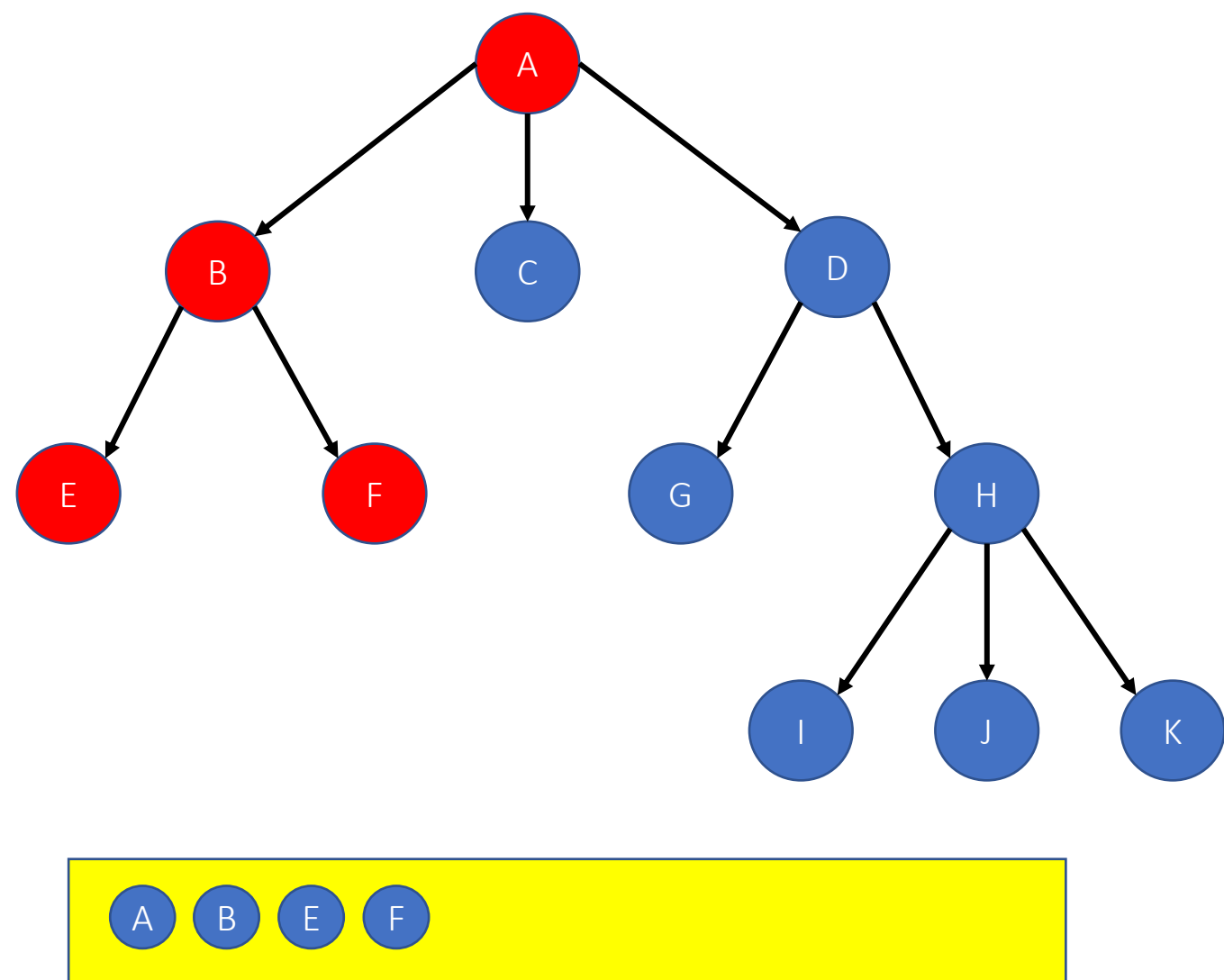
Recorriendo árboles – DFS (búsqueda en profundidad)



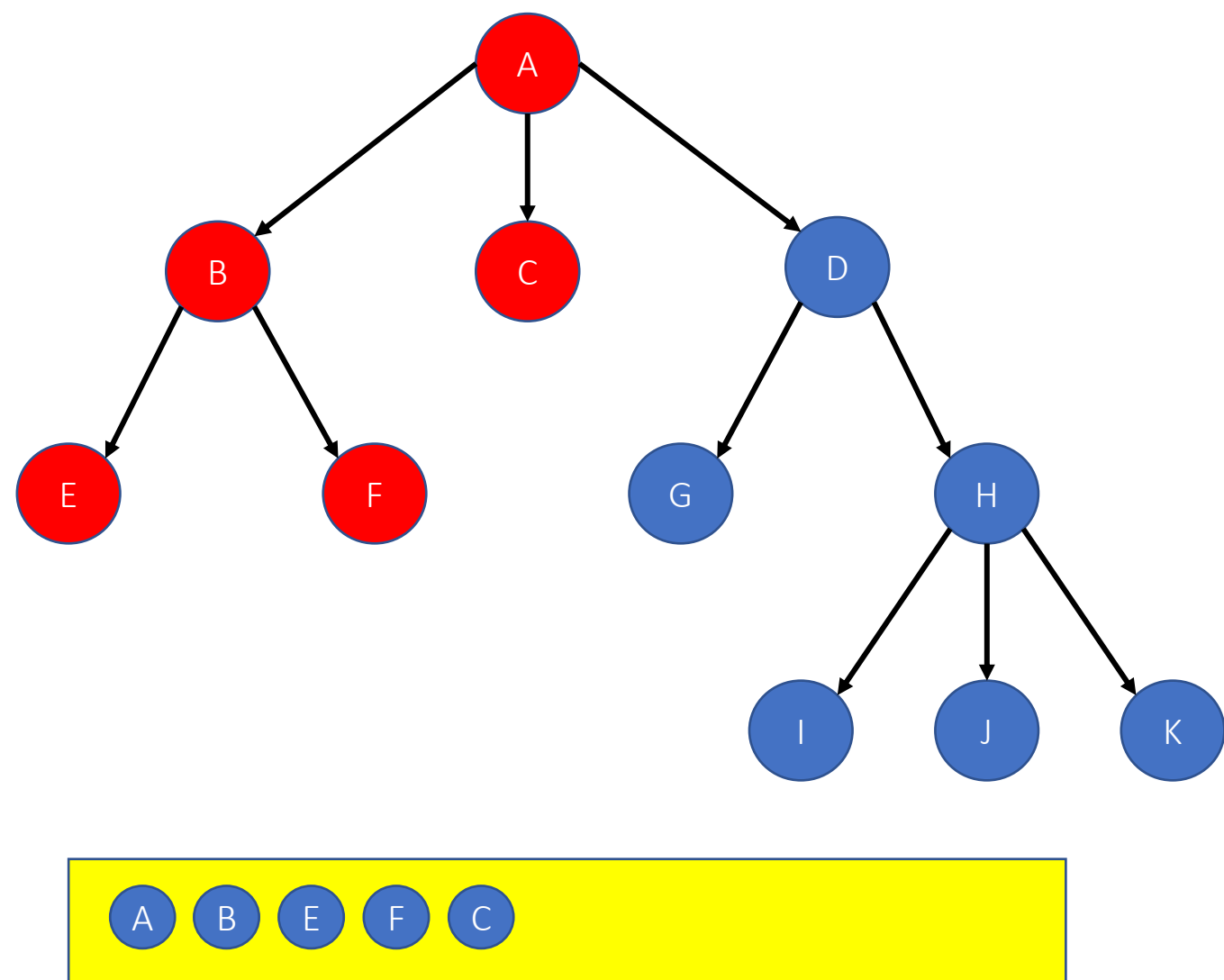
Recorriendo árboles – DFS (búsqueda en profundidad)



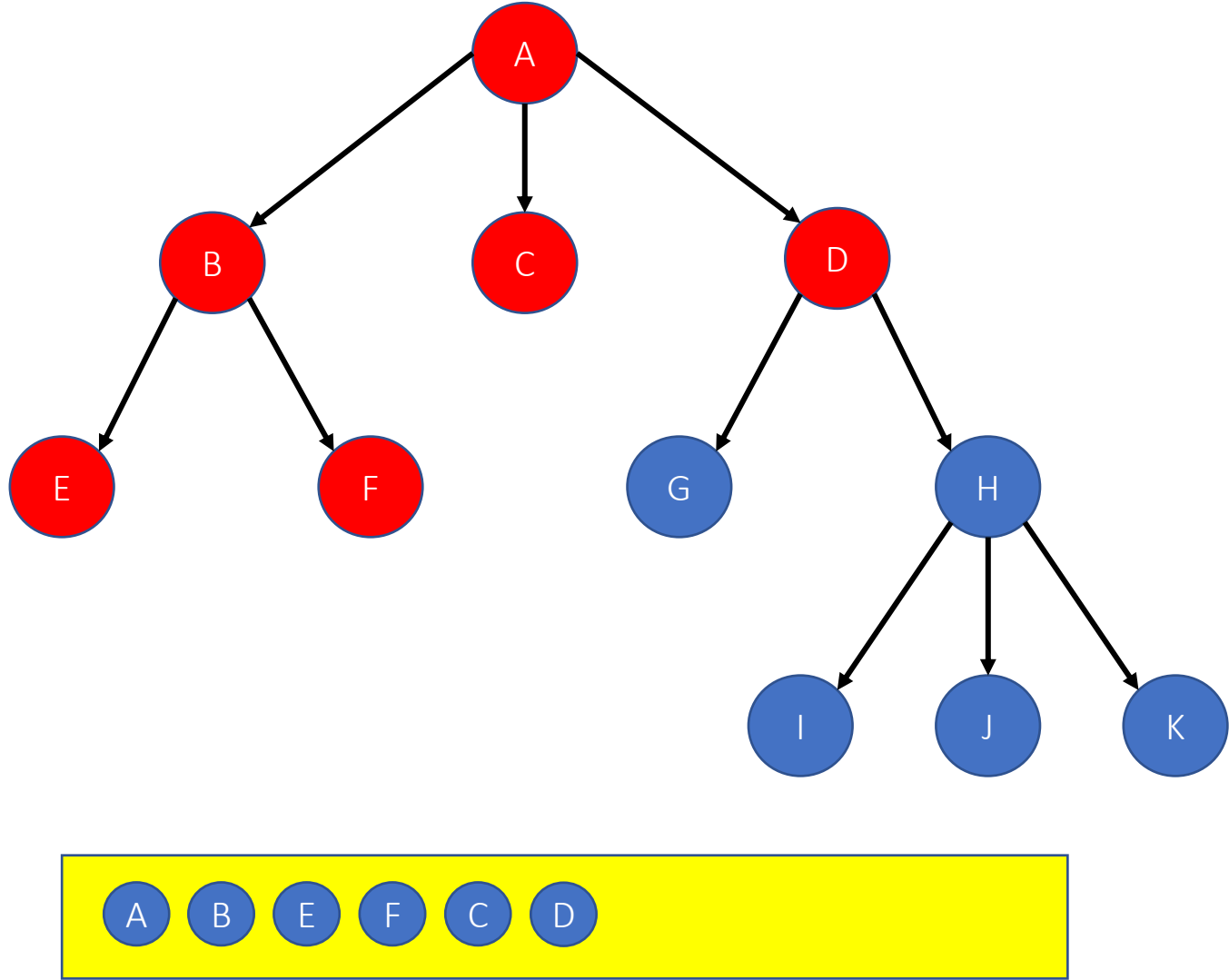
Recorriendo árboles – DFS (búsqueda en profundidad)



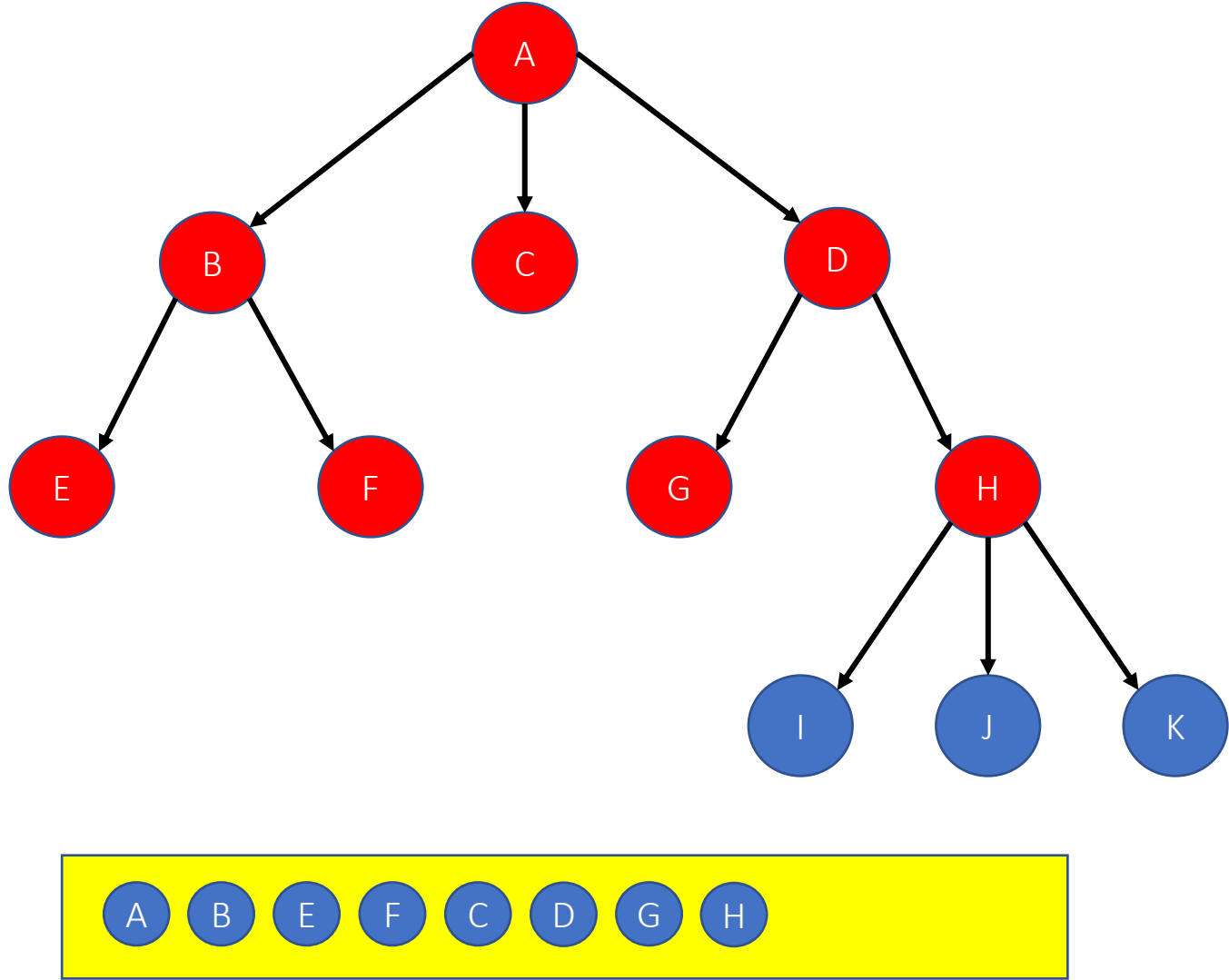
Recorriendo árboles – DFS (búsqueda en profundidad)



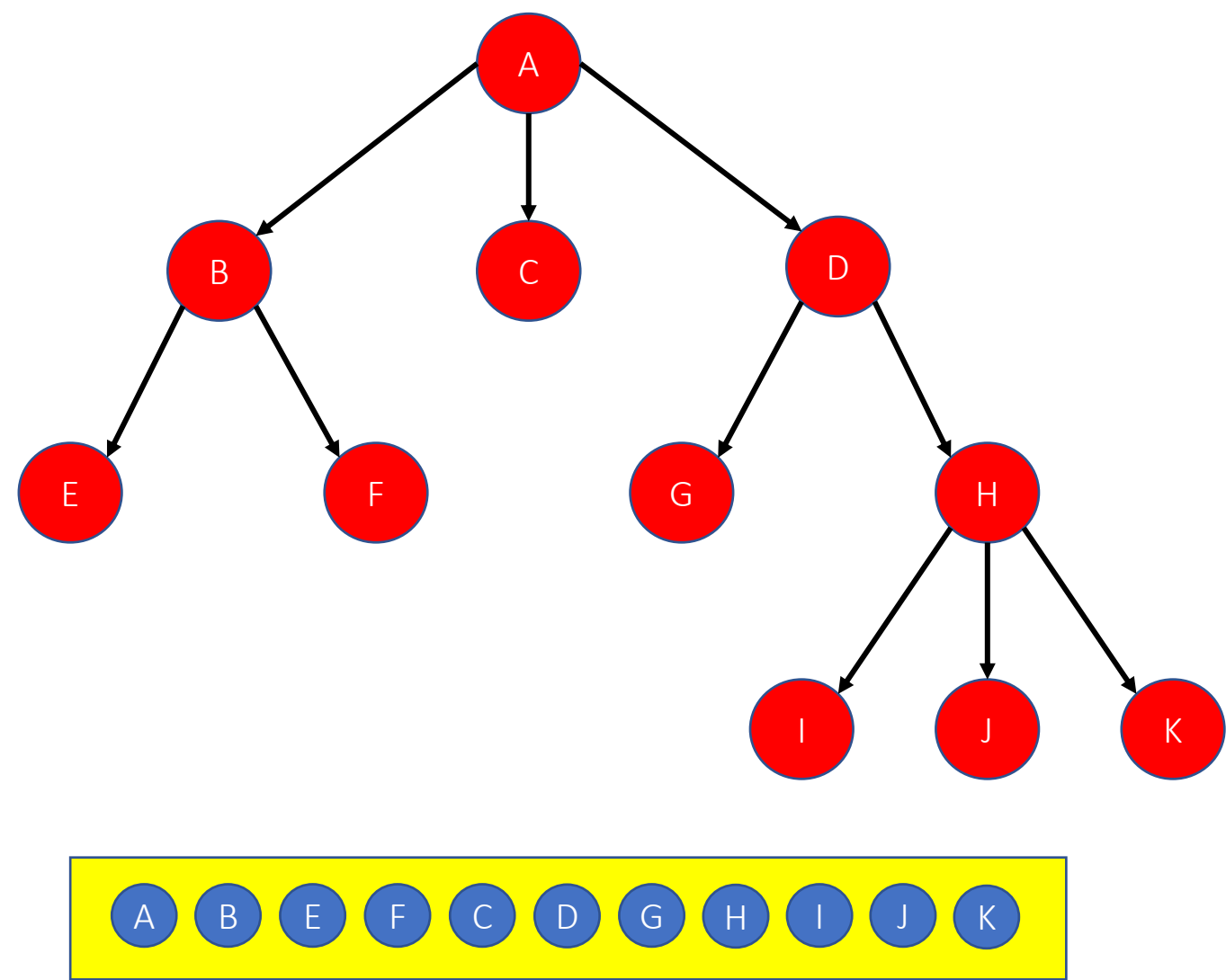
Recorriendo árboles – DFS (búsqueda en profundidad)



Recorriendo árboles – DFS (búsqueda en profundidad)



Recorriendo árboles – DFS (búsqueda en profundidad)



Un breve y somero resumen

- Las estructuras de datos corresponden a un tipo de dato especializado, **diseñado para agrupar, almacenar o acceder a la información de manera más eficiente** que un tipo de dato básico.
- La elección adecuada de la estructura de datos es fundamental para el desarrollo de un buen programa y muchas veces es la única posibilidad para solucionar un problema de forma realista.
- Pero siempre es conveniente pensar primero en una solución básica a los problemas, y luego incorporar las estructuras donde corresponda.

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación



IIC2115 - Programación como Herramienta para la Ingeniería

Estructuras de datos

Profesor: Hans Löbel