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

IIC2115 - Programación como herramienta para la ingeniería

Estructuras de datos avanzadas

Profesor: Hans Löbel

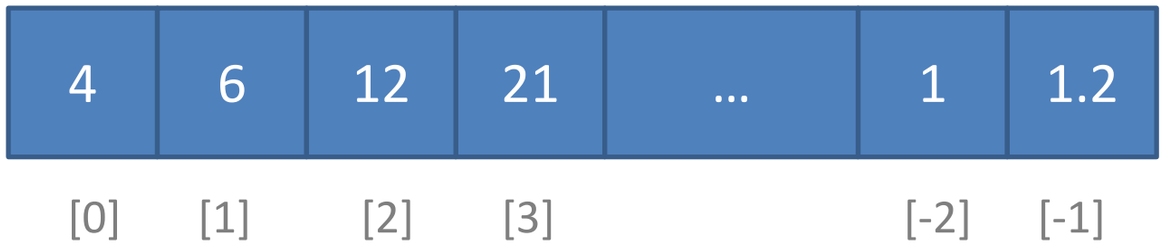
¿Qué son las estructuras de datos?

- 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 hace la diferencia entre un programador y un buen programador.

Listas, tuplas, stacks y colas

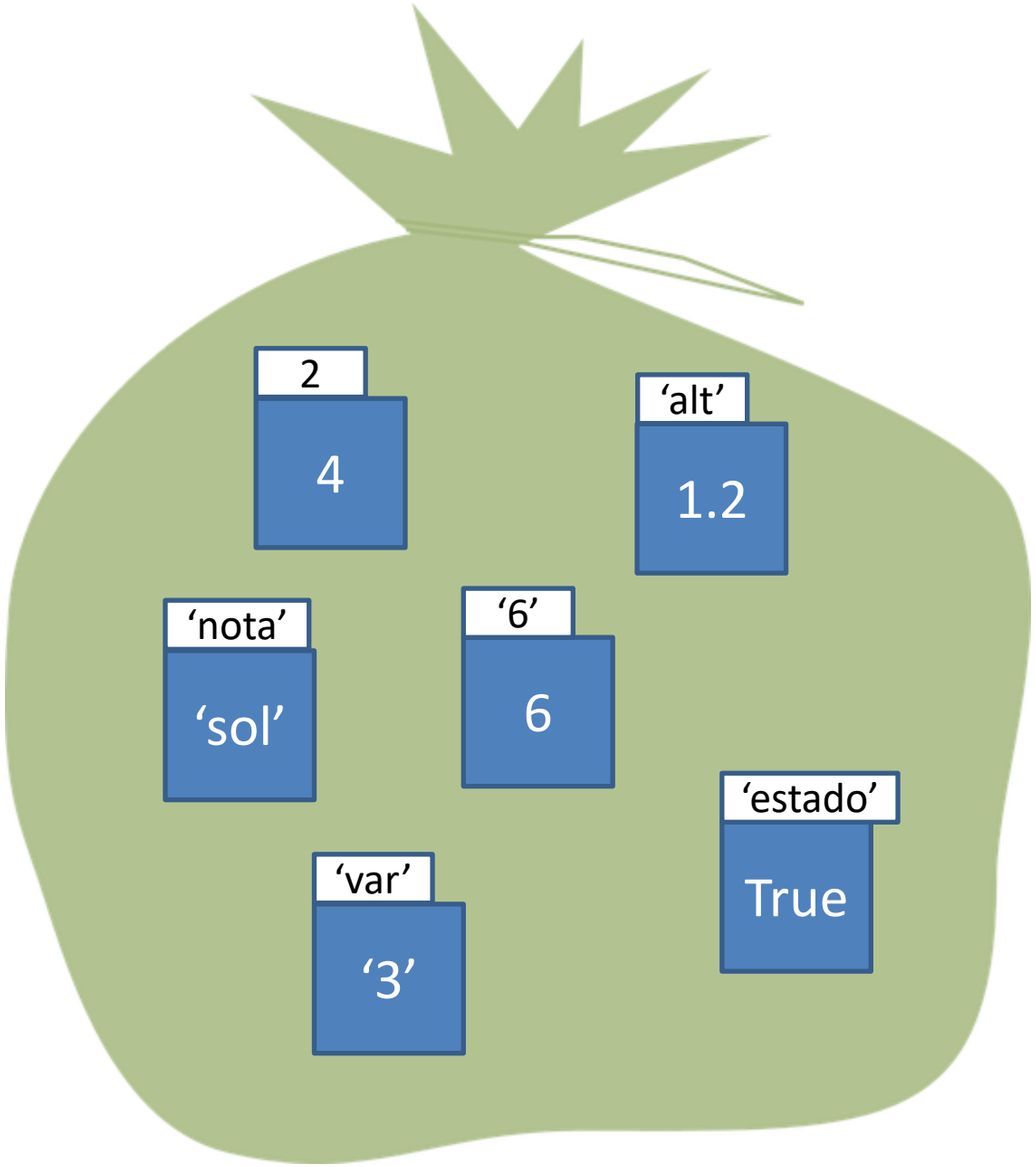
- Listas: []
- Tuplas: ()



- Stacks: LIFO (implementados con list o dequeue)
- Colas: FIFO (implementadas con dequeue)

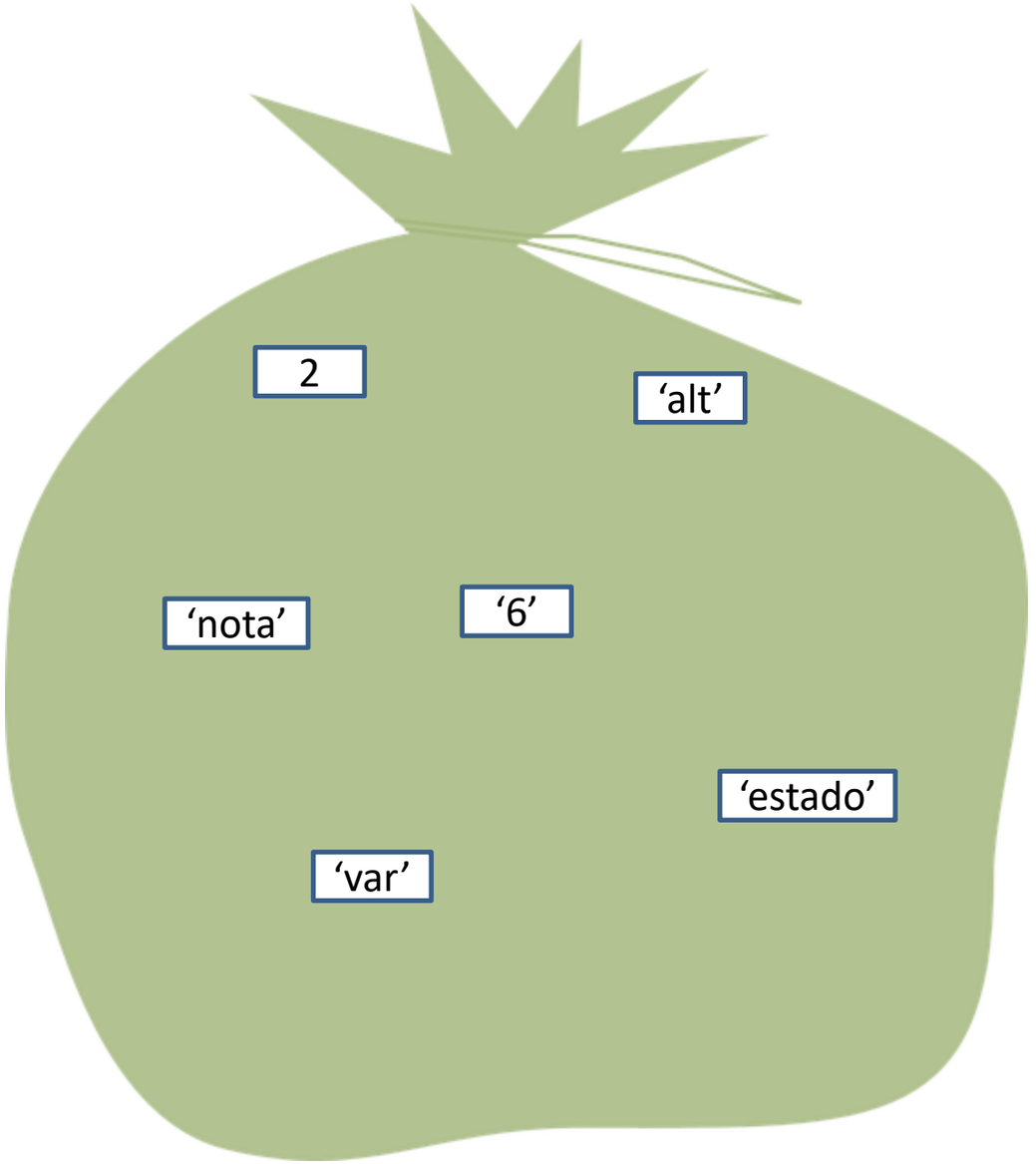
Diccionarios y Sets

- Diccionarios: {}

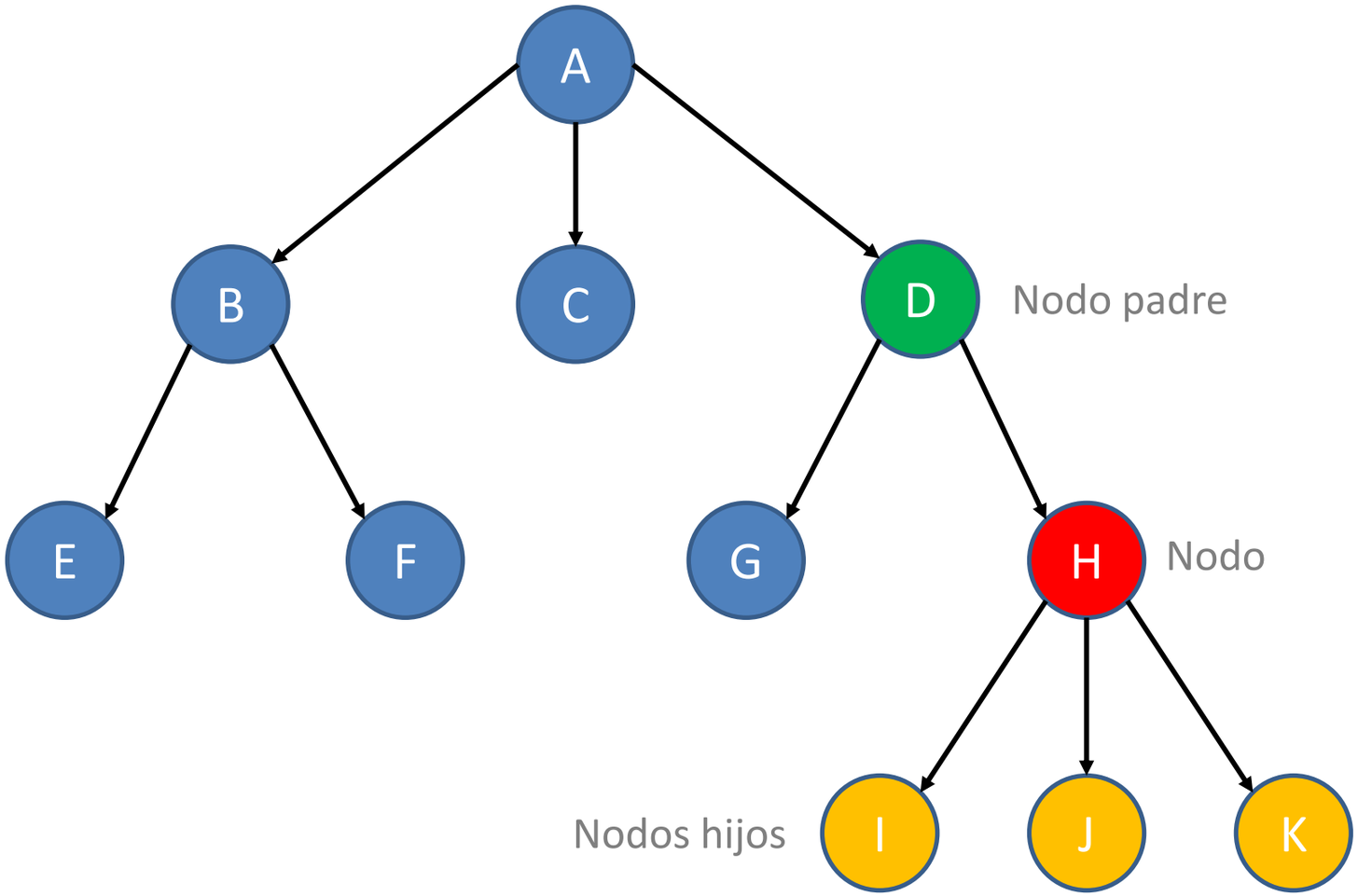


Diccionarios y Sets

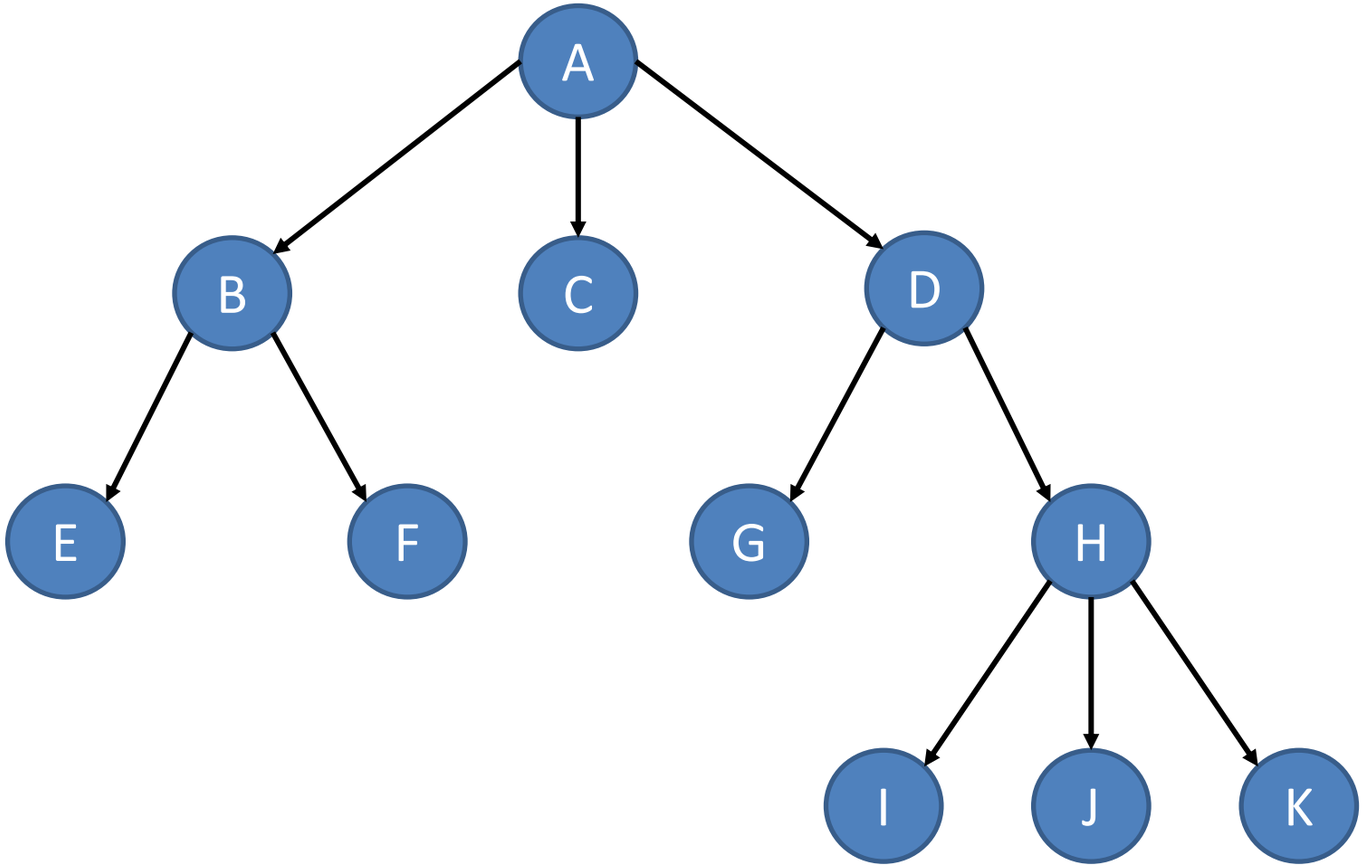
- Diccionarios: {}
- Sets: {} (no hay valores, solo llaves)



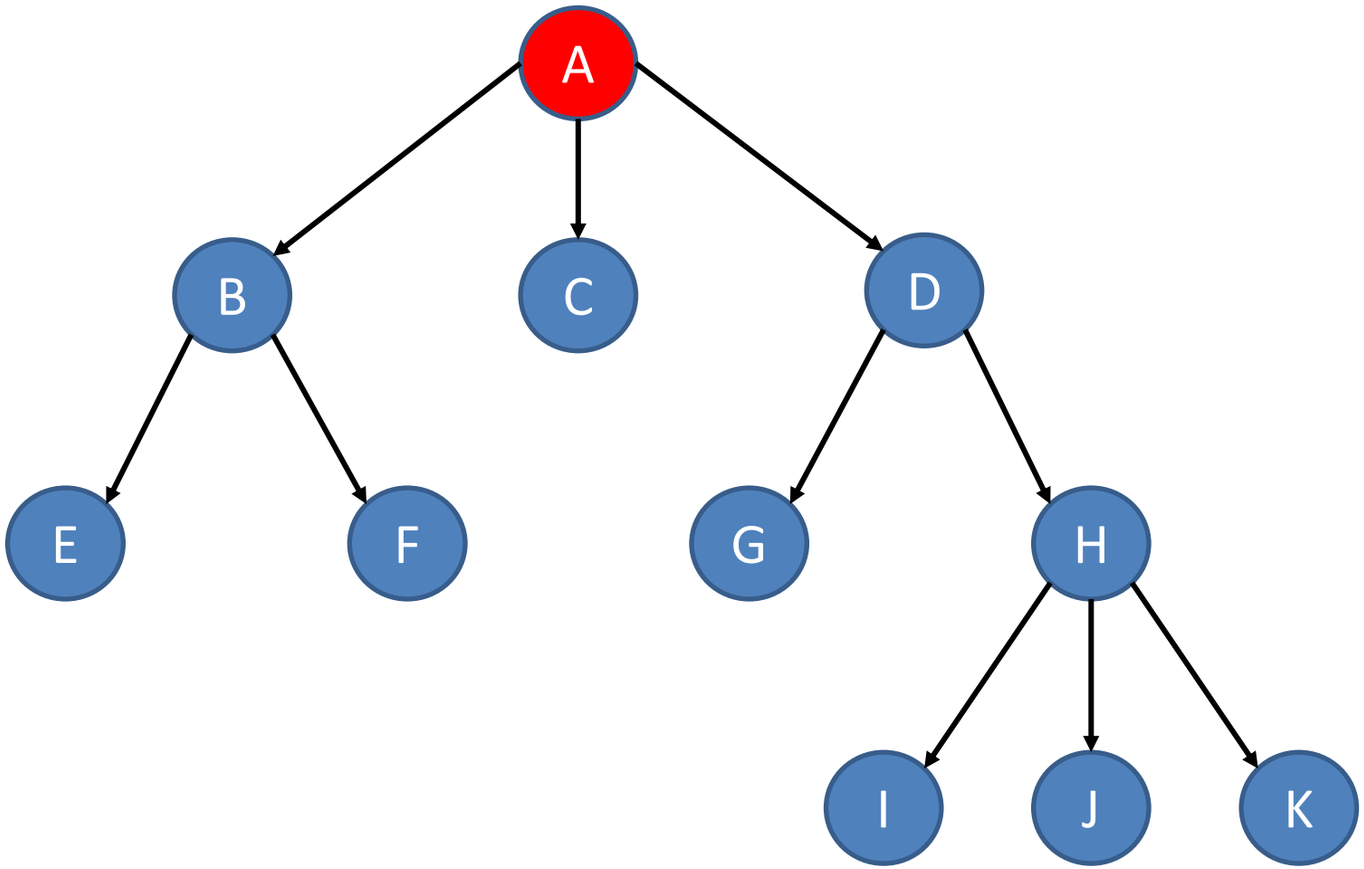
Árboles



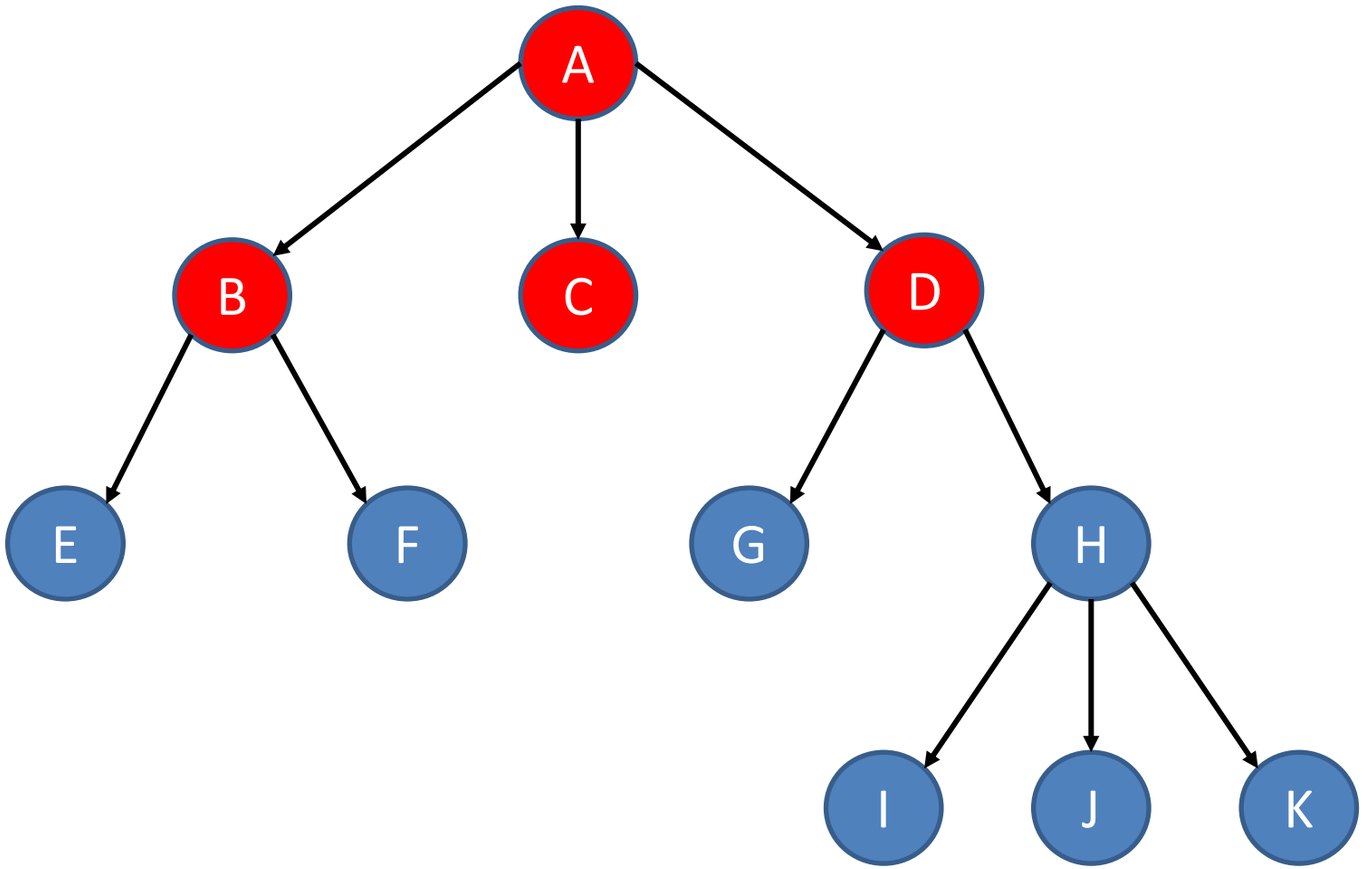
Recorriendo árboles – BFS (búsqueda en amplitud)



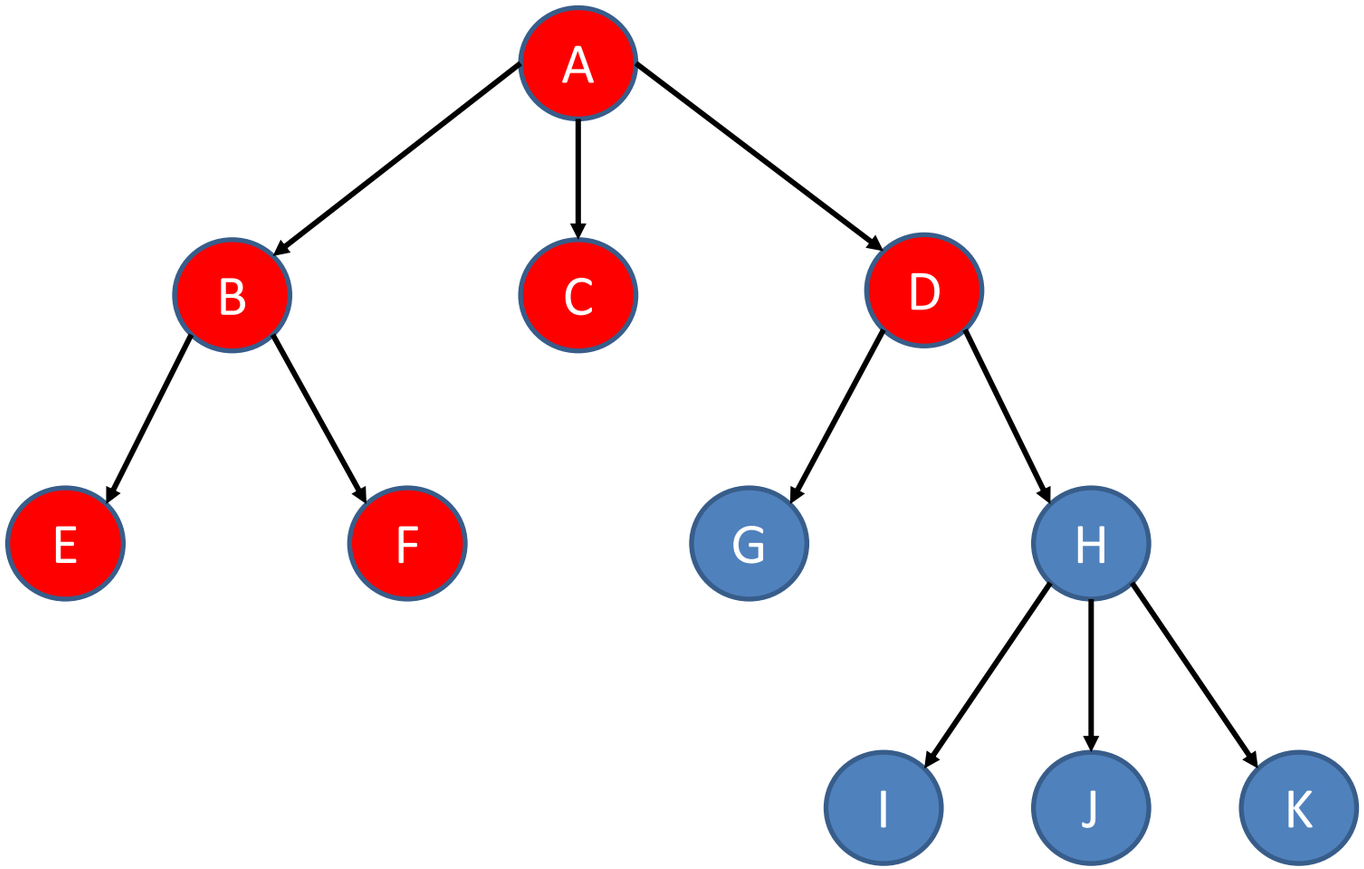
Recorriendo árboles – BFS (búsqueda en amplitud)

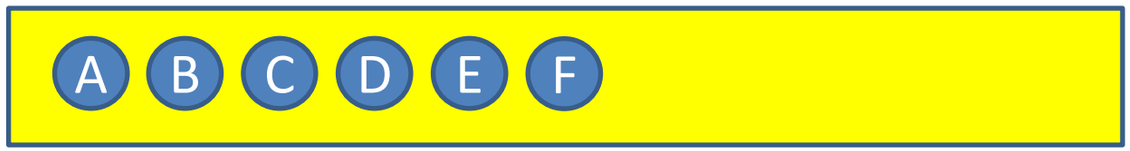


Recorriendo árboles – BFS (búsqueda en amplitud)

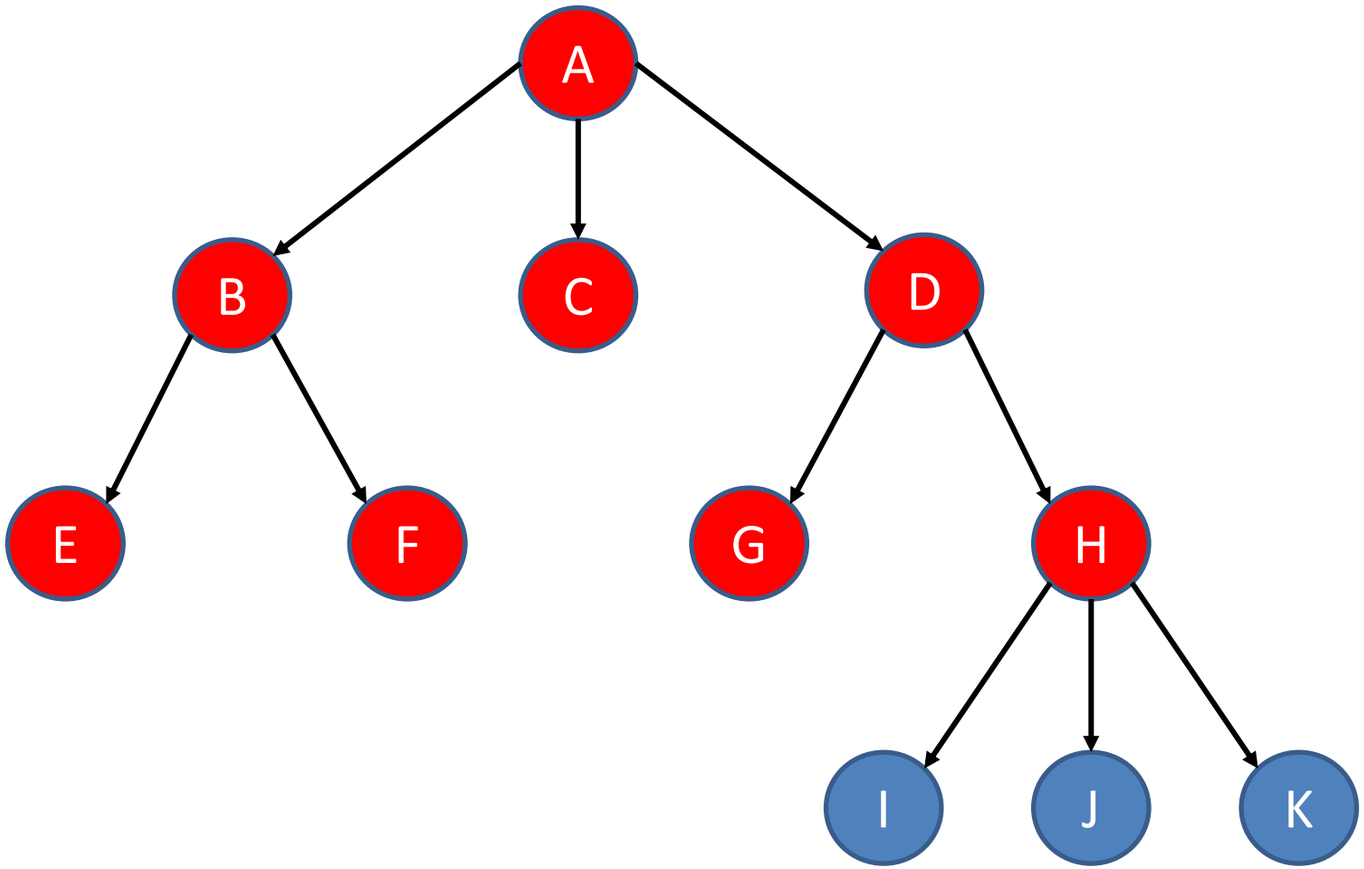


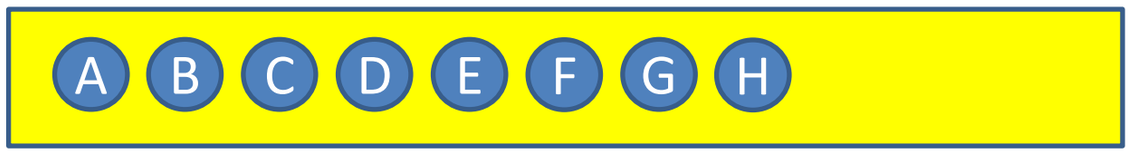
Recorriendo árboles – BFS (búsqueda en amplitud)



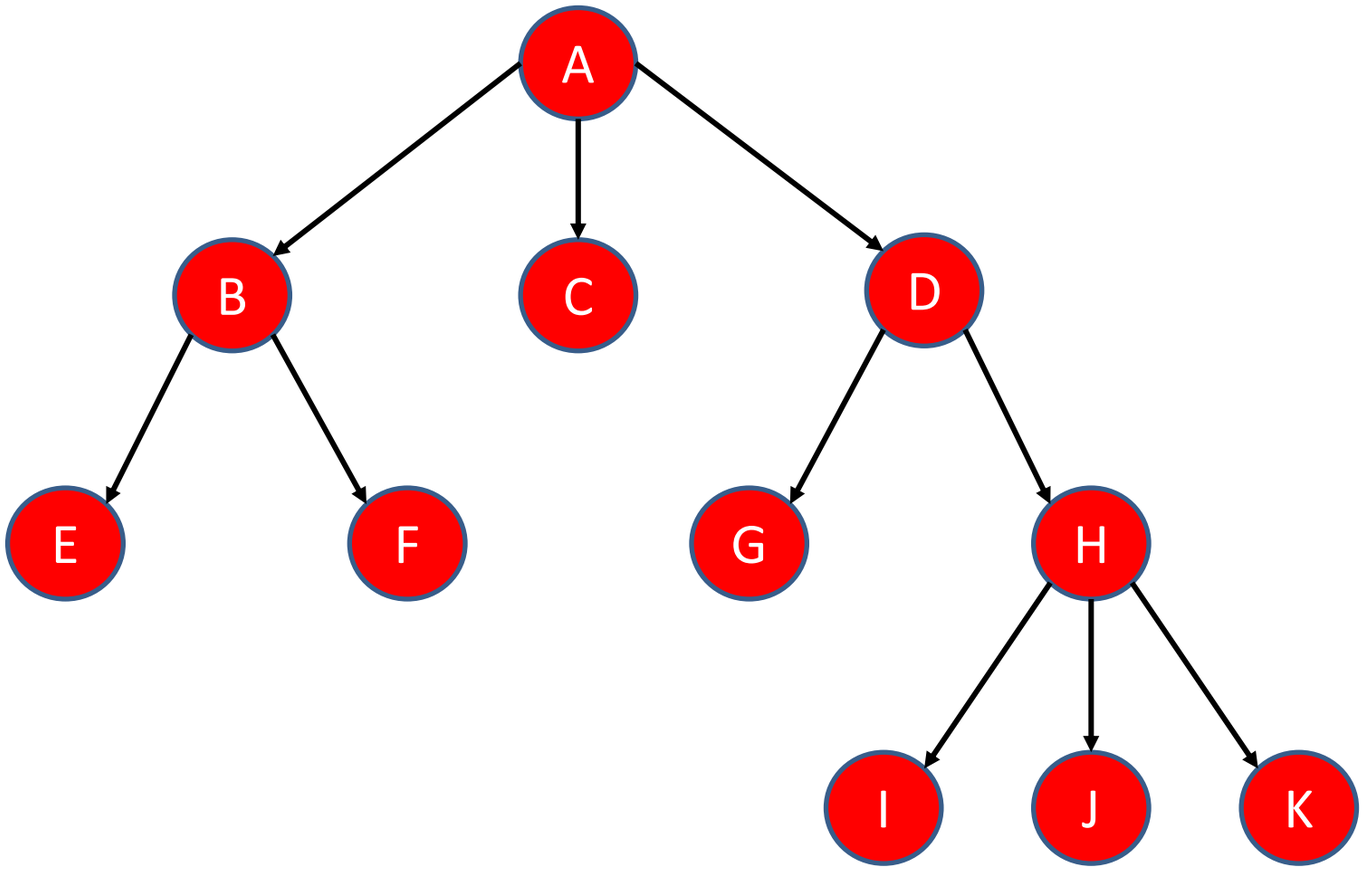


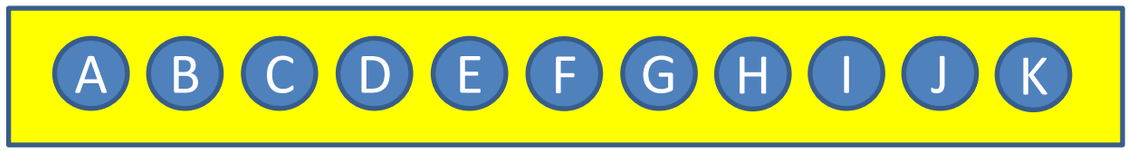
Recorriendo árboles – BFS (búsqueda en amplitud)



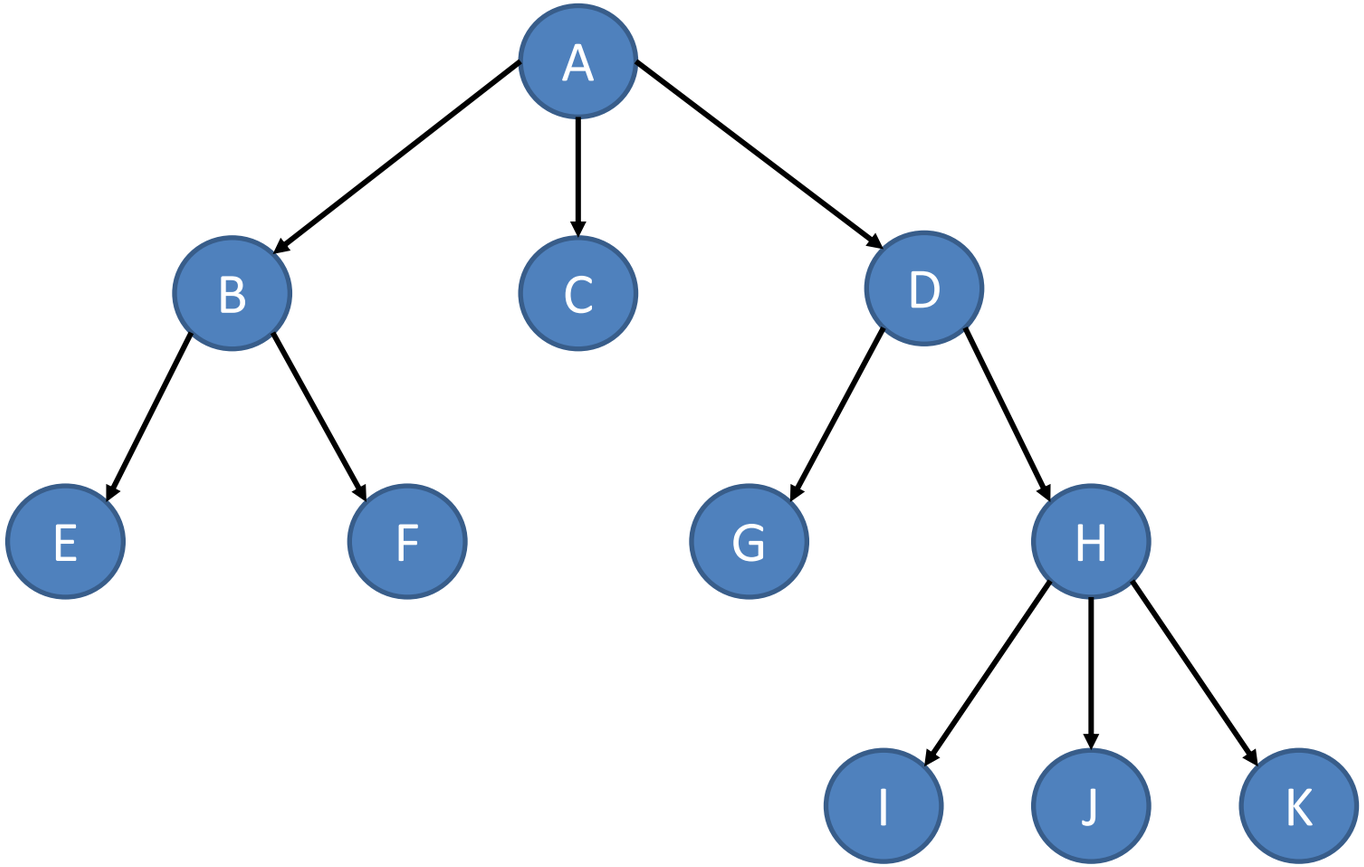


Recorriendo árboles – BFS (búsqueda en amplitud)

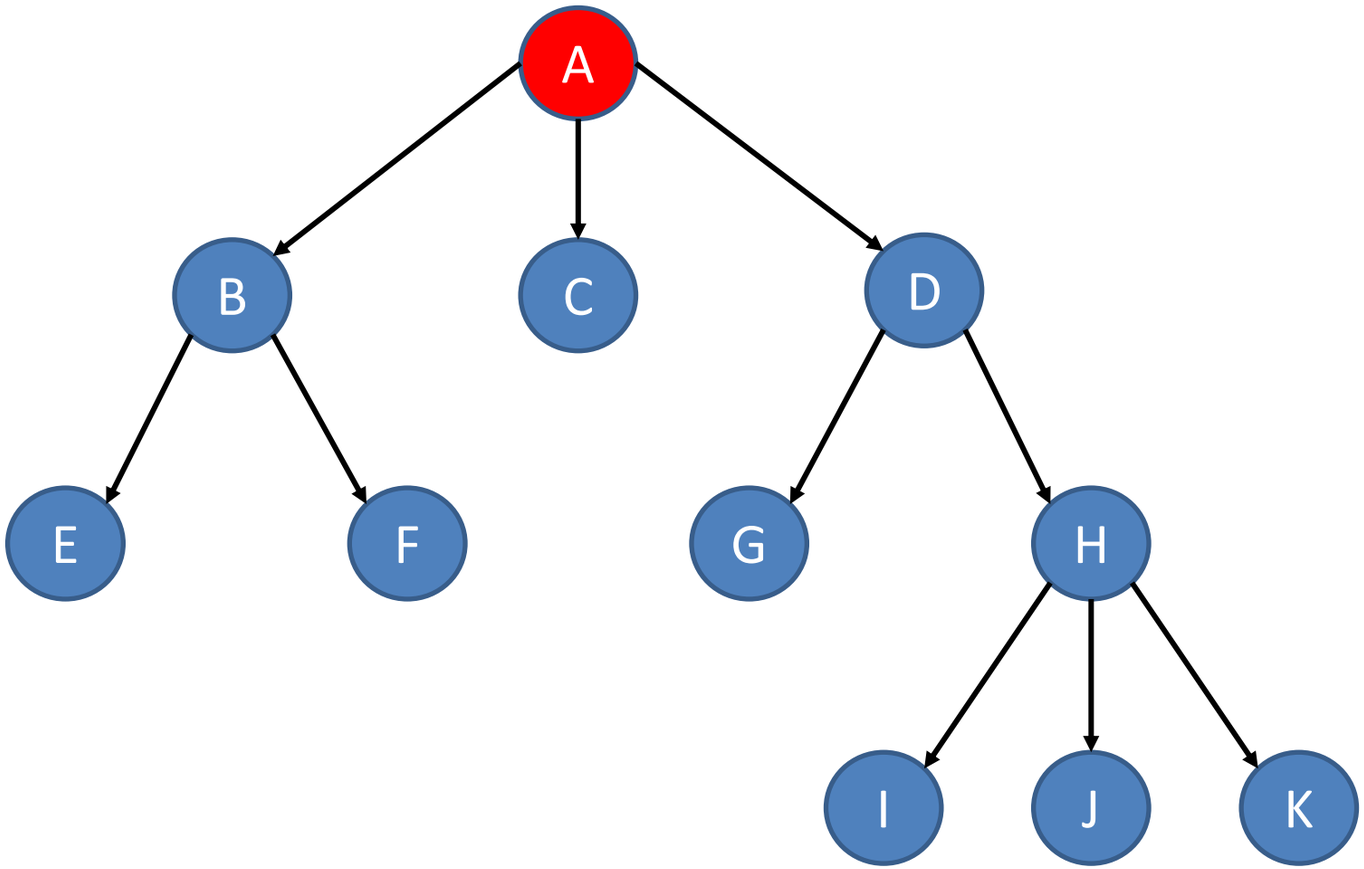




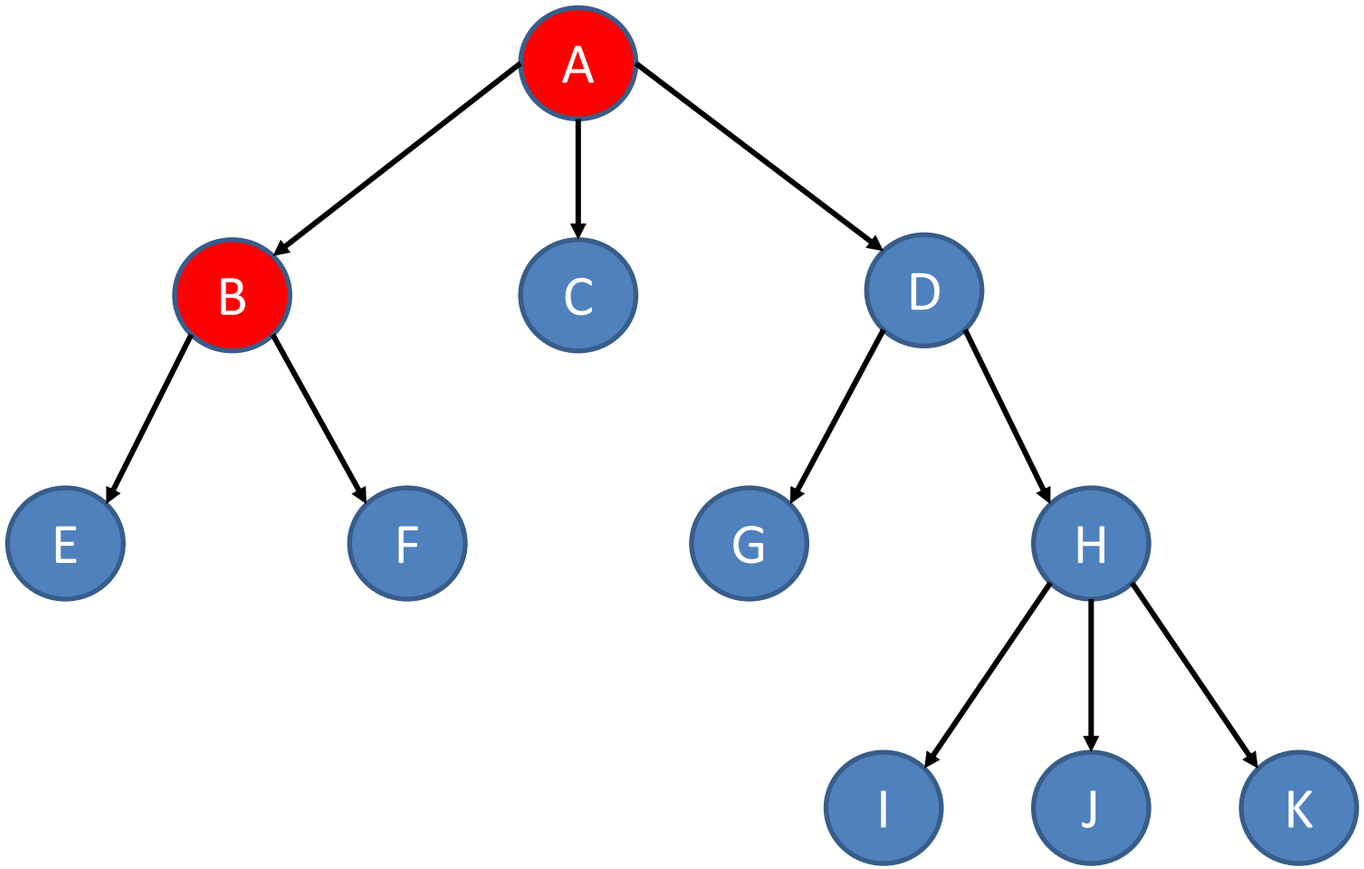
Recorriendo árboles – DFS (búsqueda en profundidad)



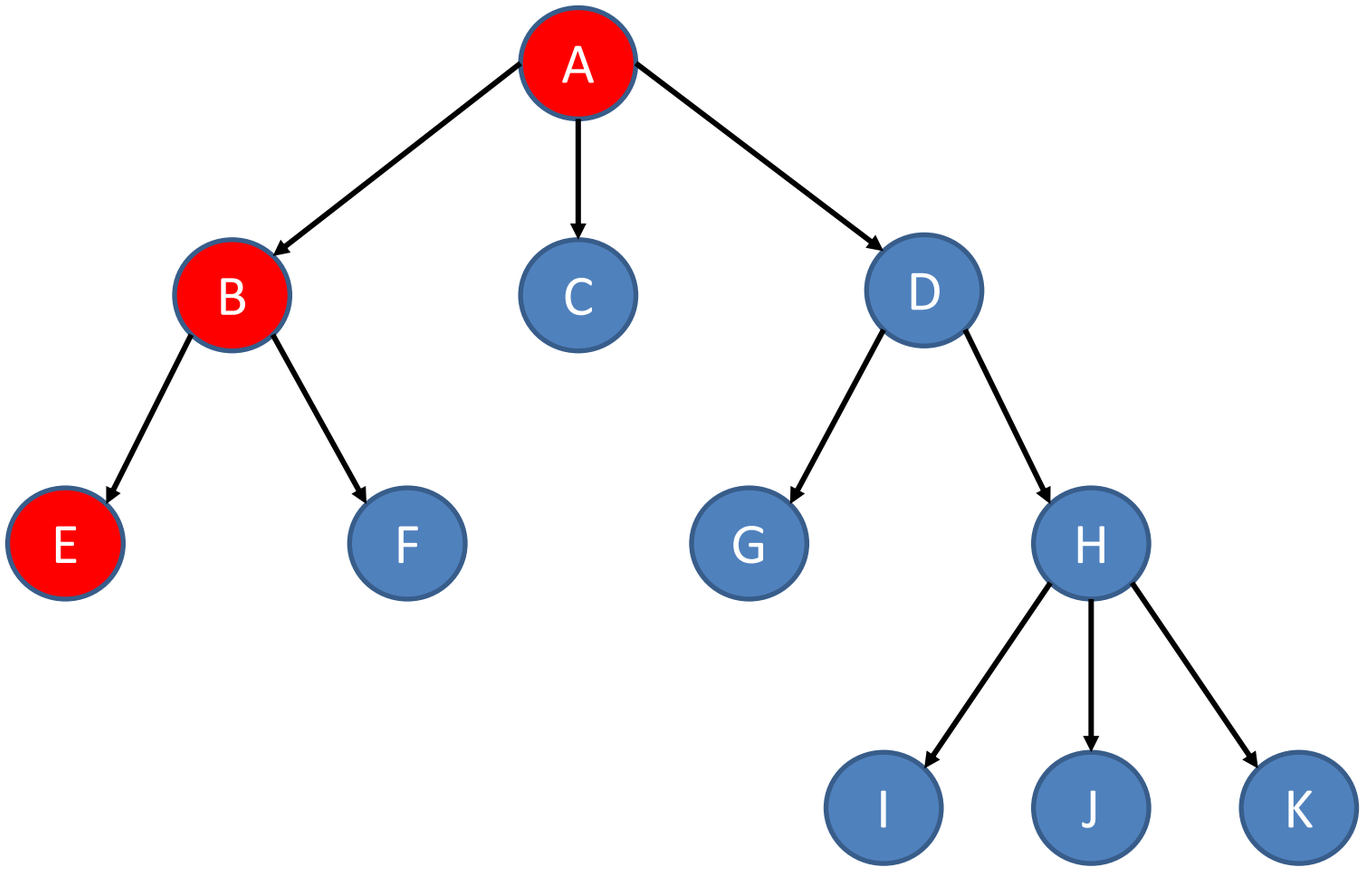
Recorriendo árboles – DFS (búsqueda en profundidad)



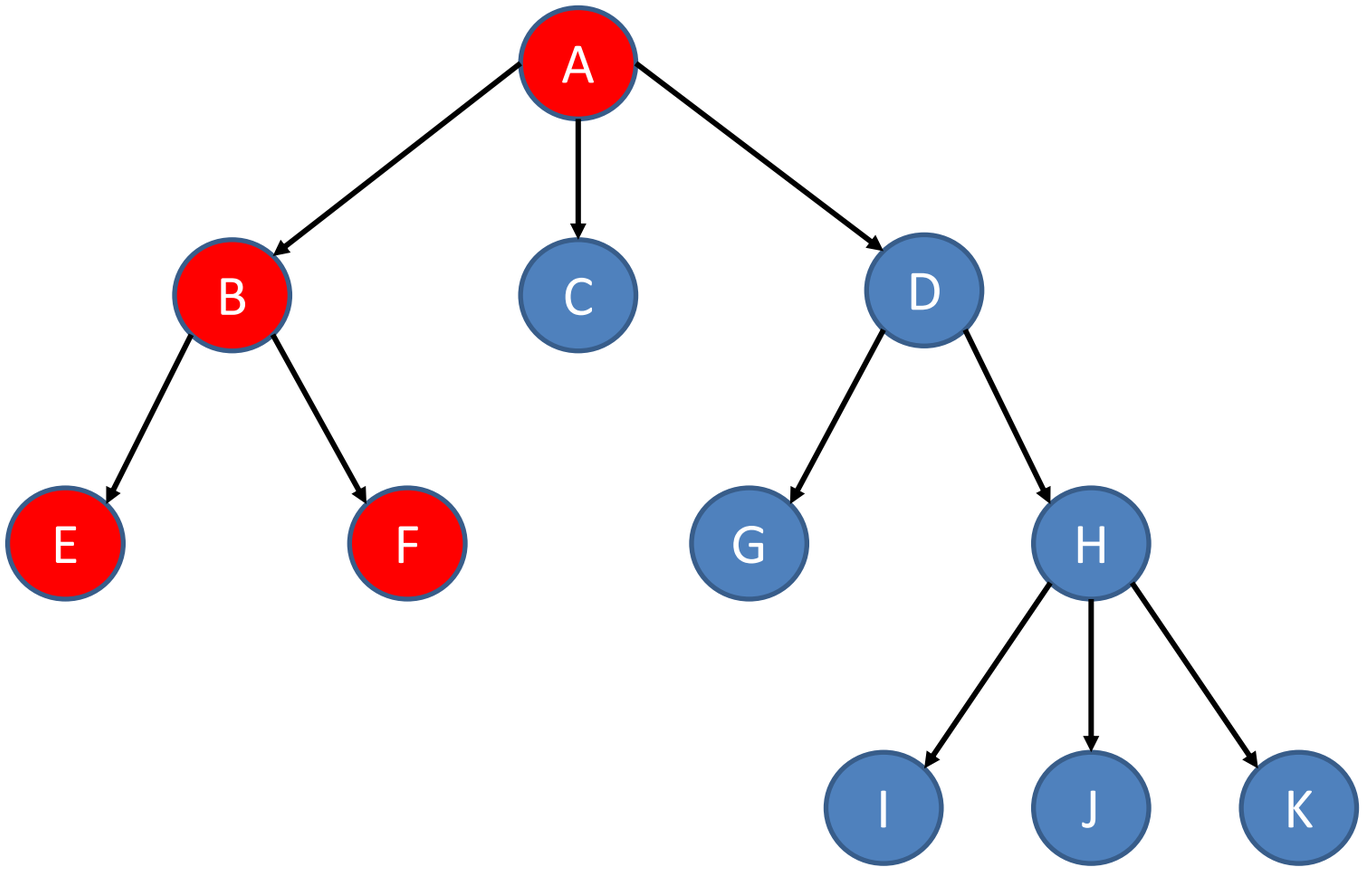
Recorriendo árboles – DFS (búsqueda en profundidad)



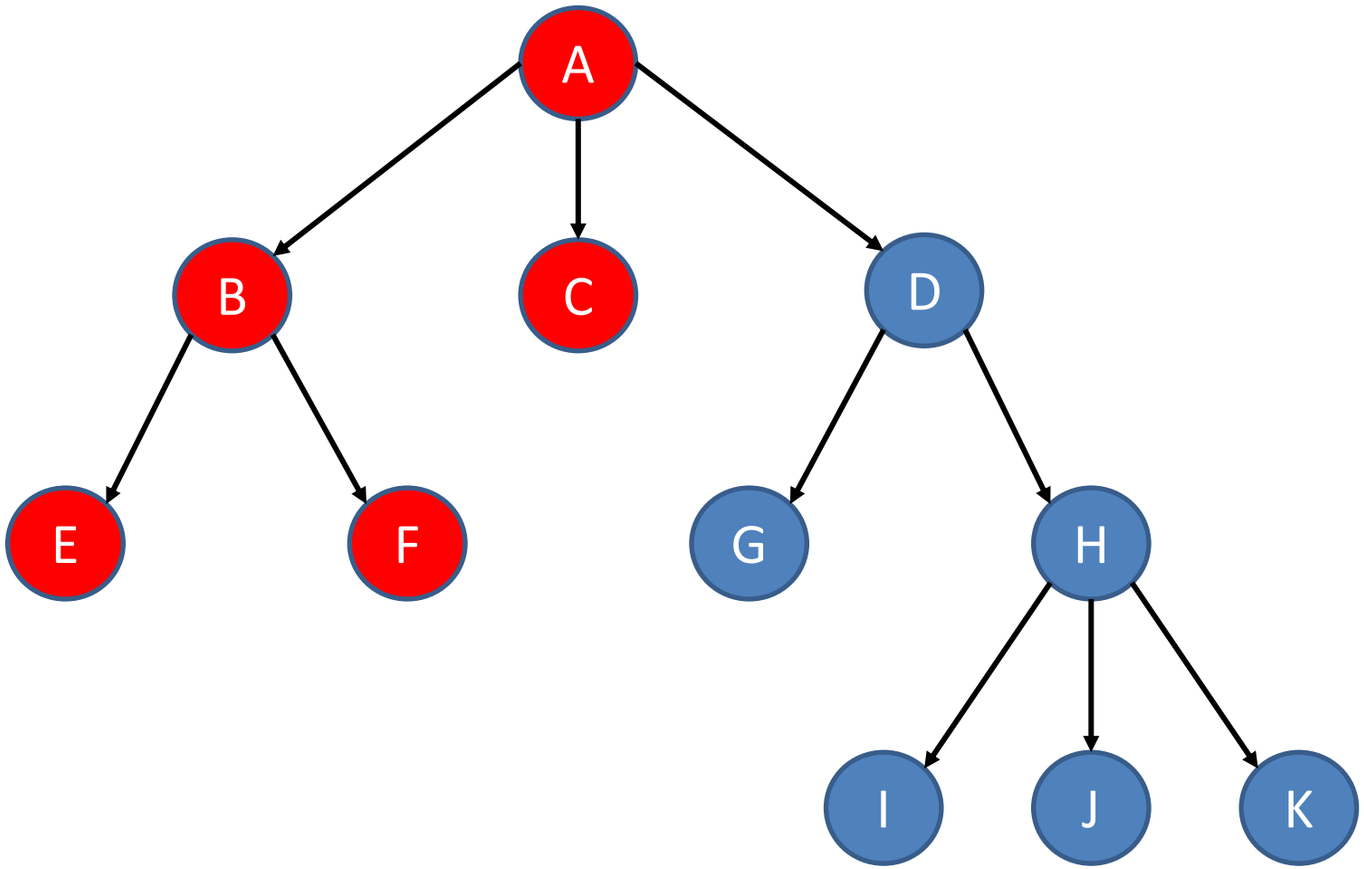
Recorriendo árboles – DFS (búsqueda en profundidad)



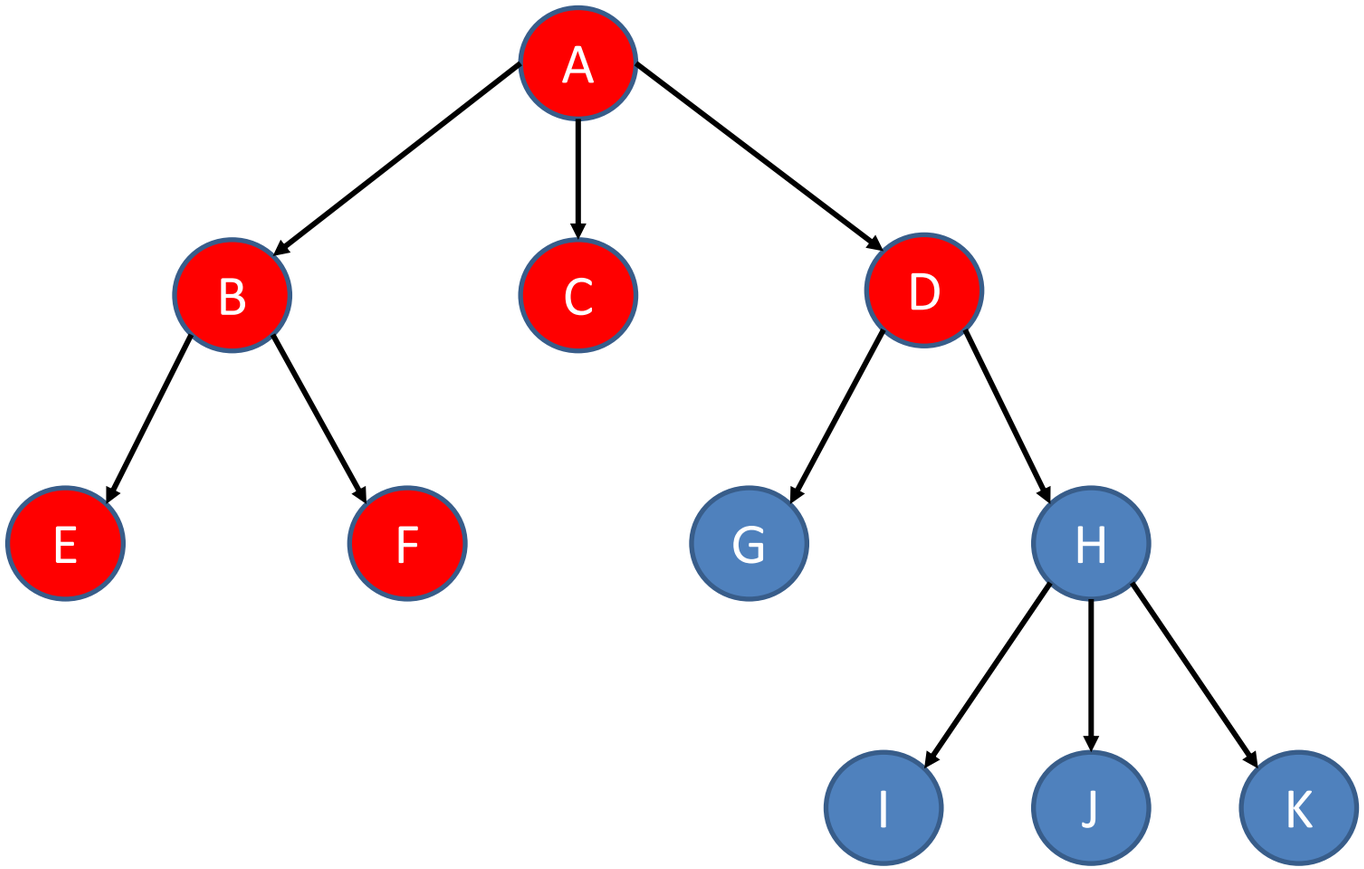
Recorriendo árboles – DFS (búsqueda en profundidad)

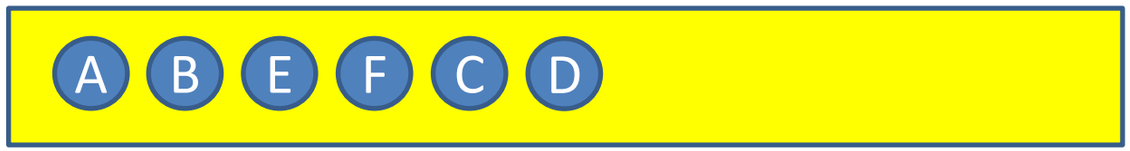


Recorriendo árboles – DFS (búsqueda en profundidad)

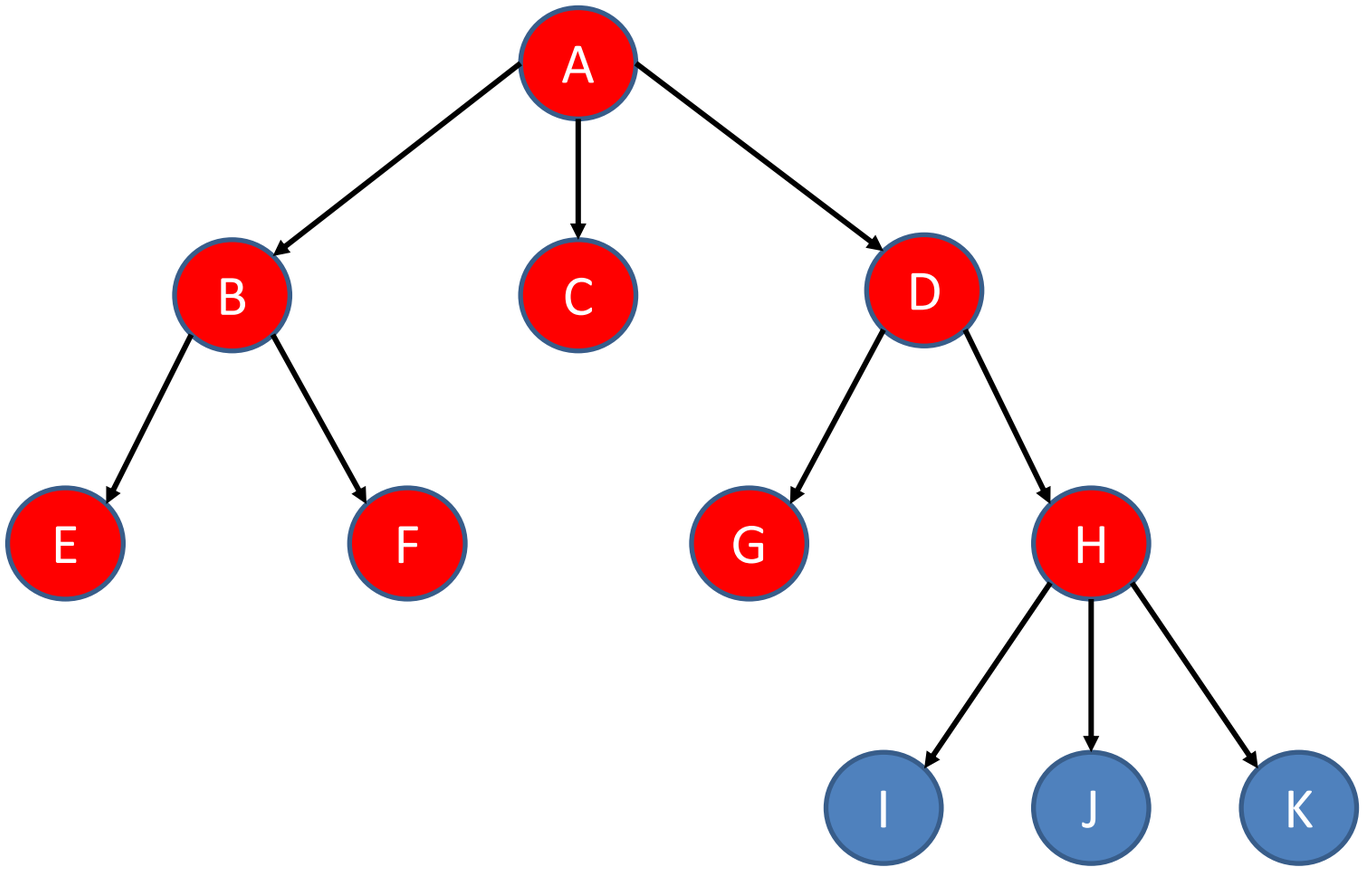


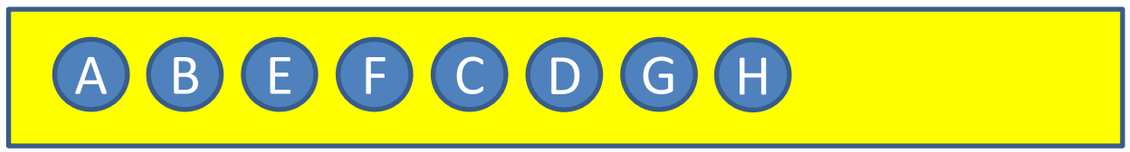
Recorriendo árboles – DFS (búsqueda en profundidad)



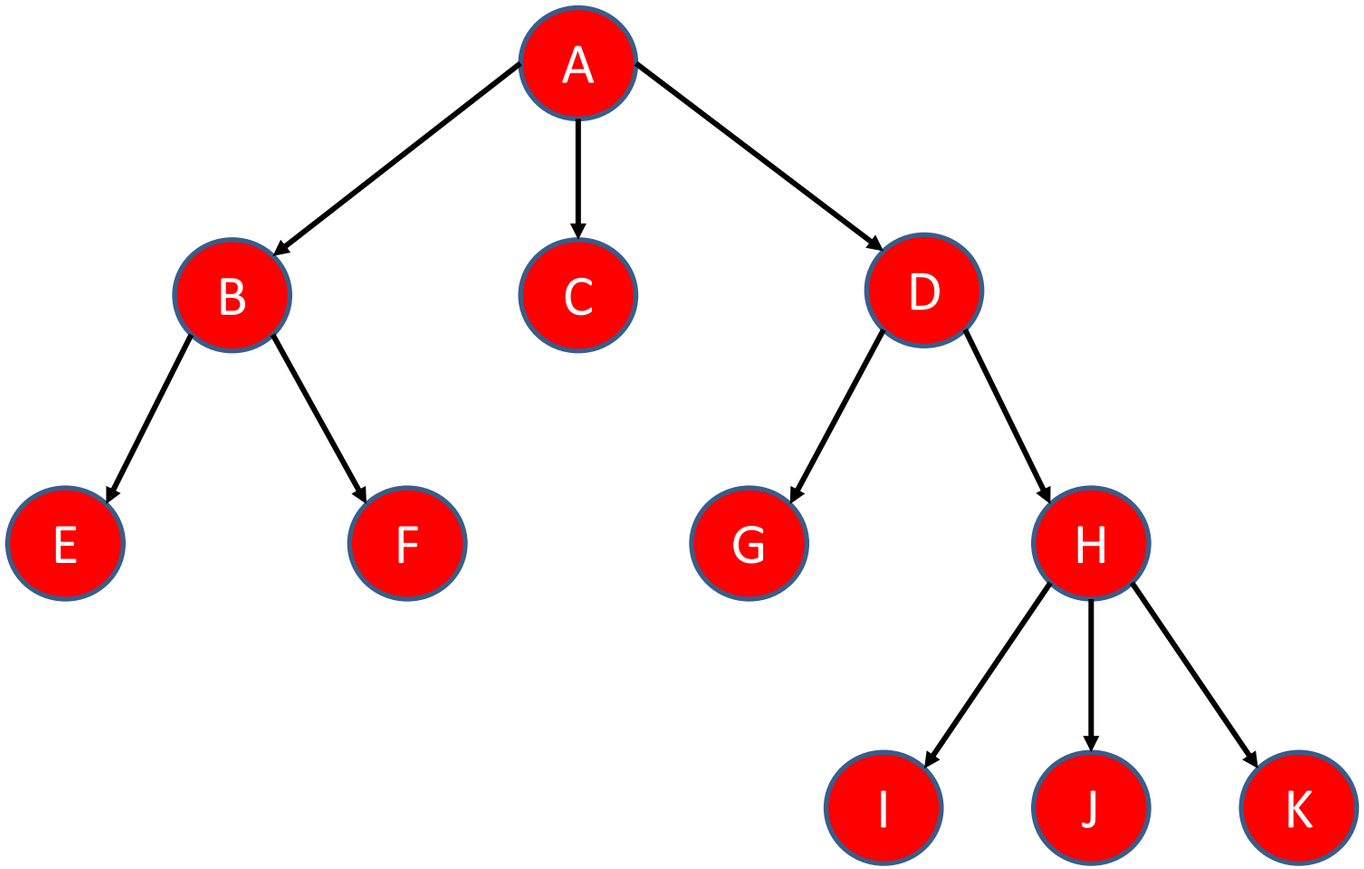


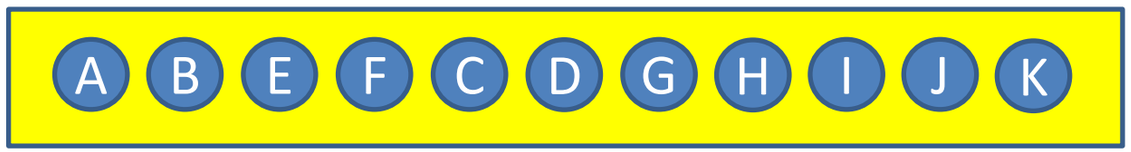
Recorriendo árboles – DFS (búsqueda en profundidad)





Recorriendo árboles – DFS (búsqueda en profundidad)

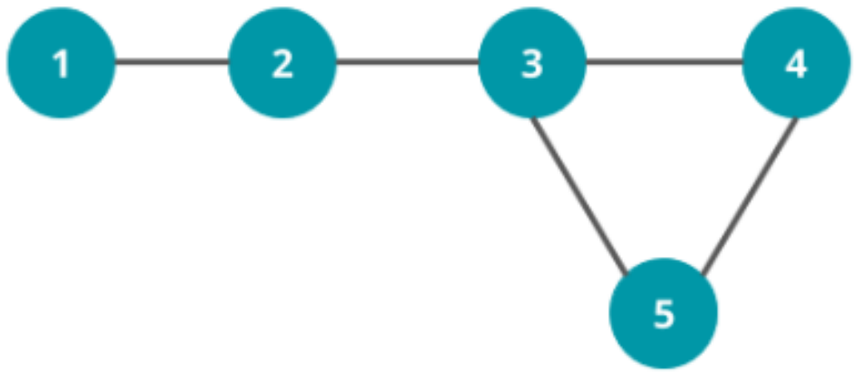




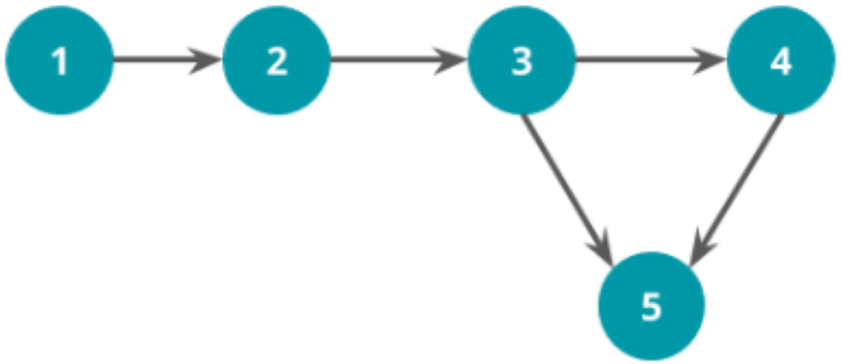
Grafos

- Representables a través matrices de adyacencia o incidencia.
- En ambos casos se pueden codificar los pesos de los arcos.

Grafo no dirigido



Grafo dirigido



¿Cómo conviene estudiar esta materia?

Pensar, pensar, pensar y luego al código

- Entre más problemas resuelvan, más fácil será la resolución de futuros problemas
- Es importante entender completamente el problema antes de escribir algo de código.
- Funciona bien usar ejemplos pequeños para facilitar la comprensión.
- Una vez que se tiene clara una primera versión (básica o fuerza bruta, da lo mismo), empezar a programar.
- Luego, revisitar la solución cuantas veces sea necesario, identificando los cuellos de botella.

Un ejemplo práctico para terminar

“Dado un string que utiliza los parentesis:

( ) [ ] { }

determine si se encuentra balanceado o no”

1. Entender el enunciado
2. Darnos ejemplo sencillos para entender la mecánica
3. Pensar formas de abordar el problema
4. Programo
5. Puedo mejorarlo?
6. Sigo programando

Si recibieramos el texto ‘(){{[]})’ debiesemos retornar True, mientras que con ‘([)]’ o ‘(([]}’ False

Stack

Stack

Stack

Stack

Stack

Stack

Stack

Si el stack estuviese vacío, el retorno sería False

Stack

Si el tope del stack corresponde al opuesto del que estamos analizando, lo sacamos y seguimos revisando, en caso contrario retornamos False

Stack

Stack

Stack

Seguimos iterando y si al finalizar el proceso el stack está vacío, retornamos True, en caso contrario, False

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

IIC2115 - Programación como herramienta para la ingeniería

Estructuras de datos avanzadas

Profesor: Hans Löbel