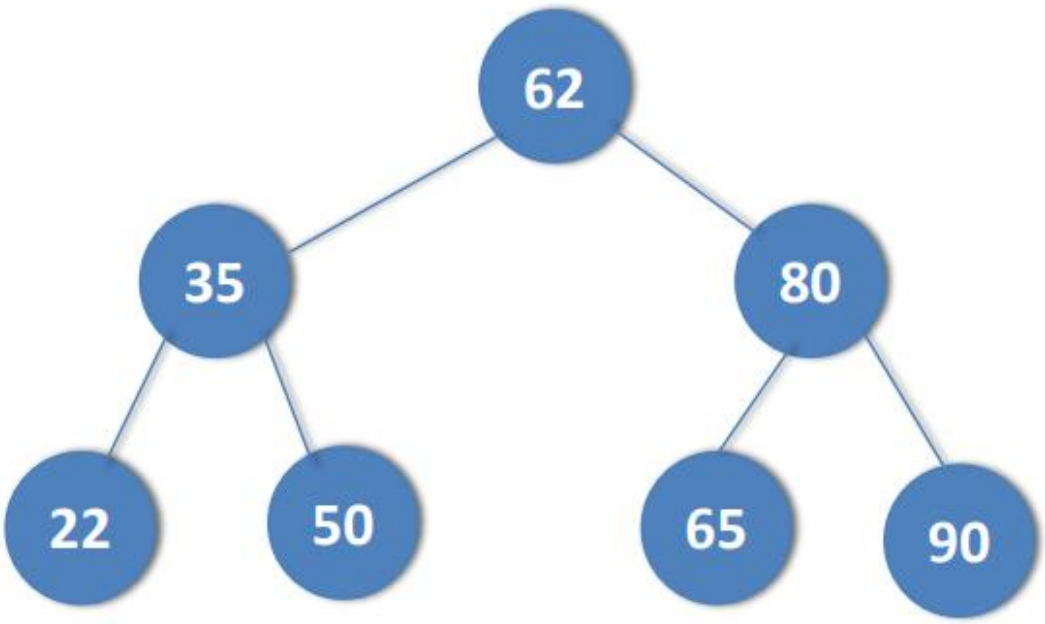
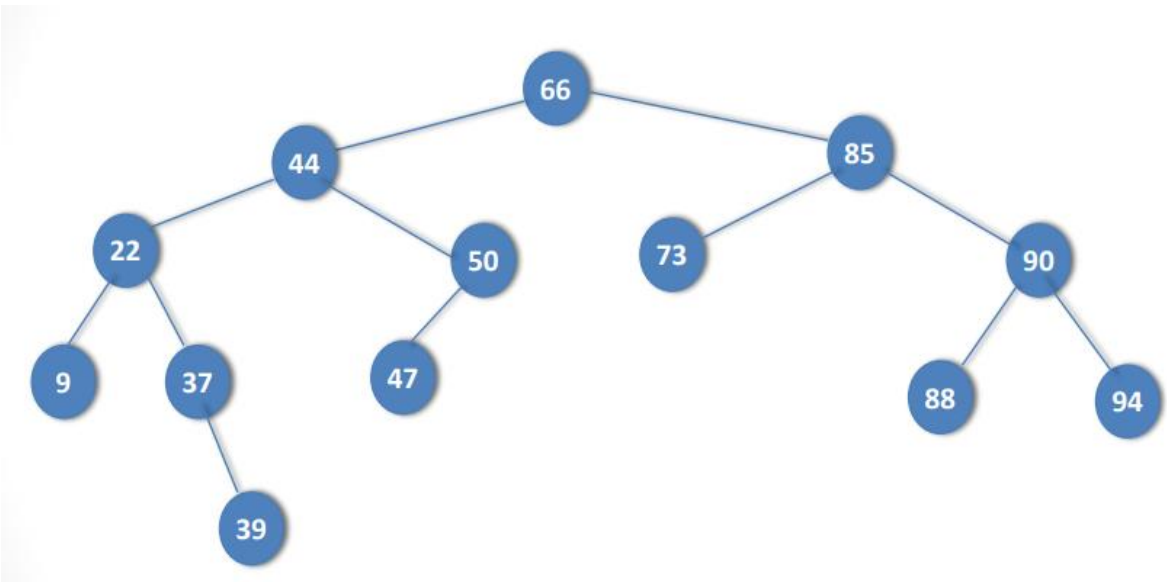
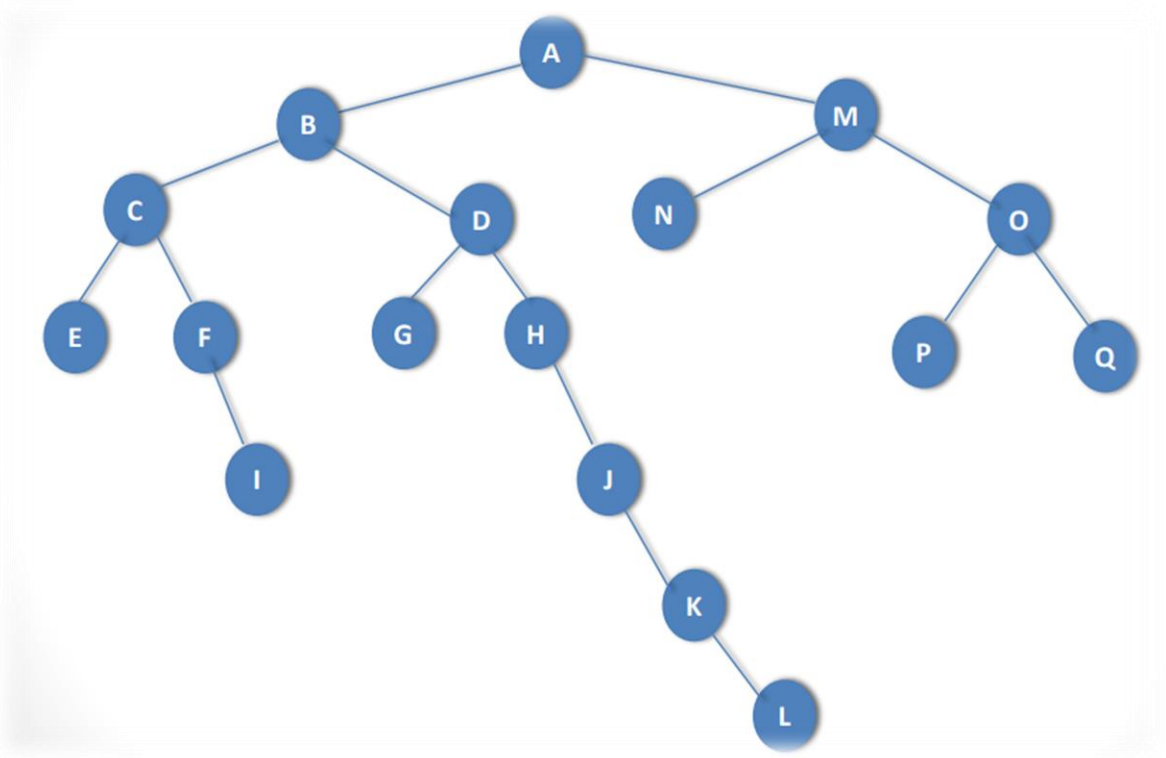
1. Determinar recorrido PreOrden, InOrden y PosOrden.



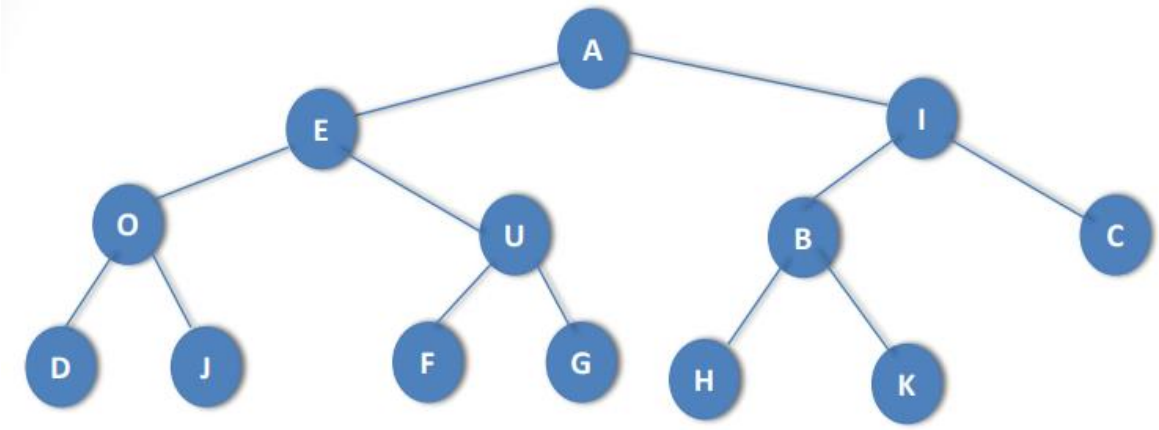
2. Determinar recorrido PreOrden, InOrden y PosOrden.



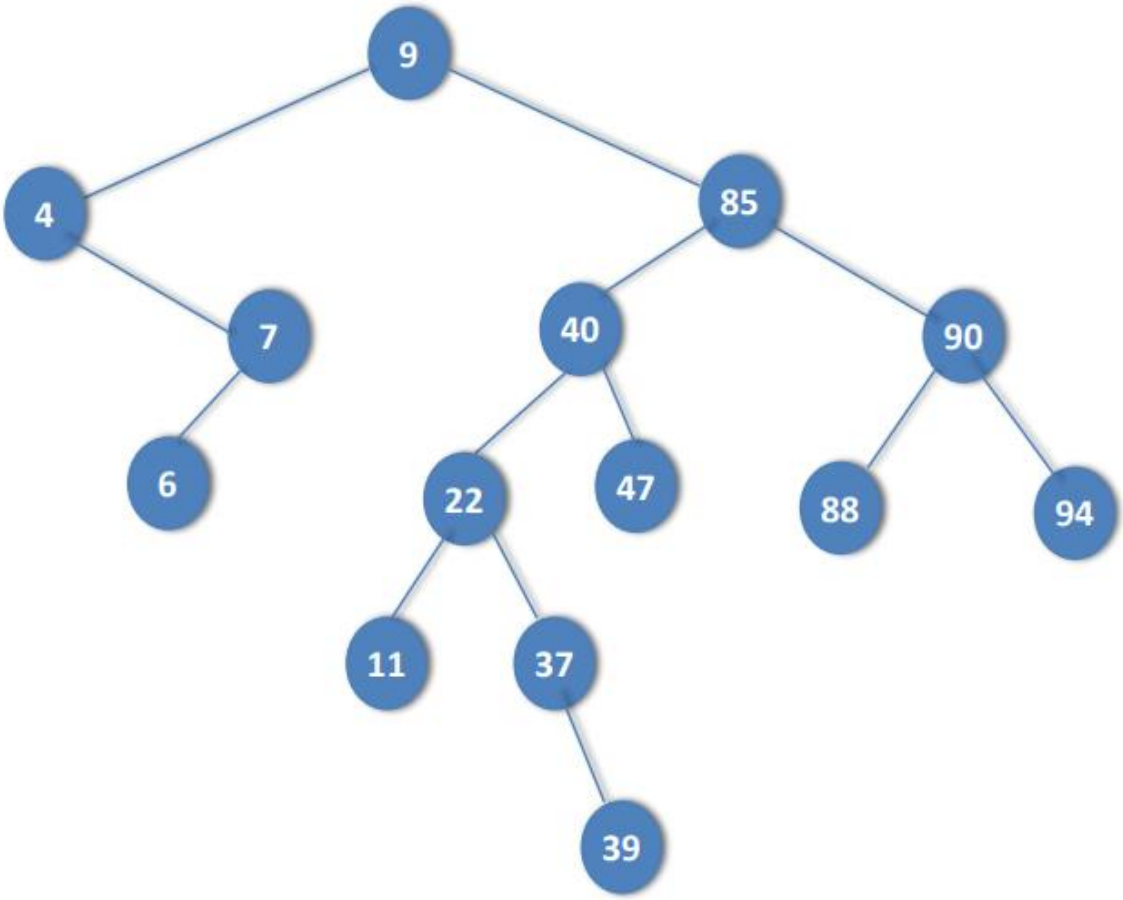
3. Determinar recorrido PreOrden, InOrden y PosOrden.



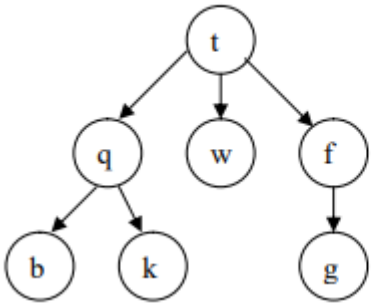
4. Determinar recorrido PreOrden, InOrden y PosOrden.

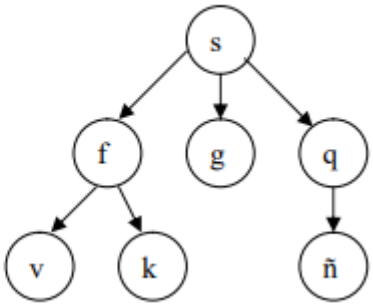


5. Determinar recorrido PreOrden, InOrden y PosOrden.

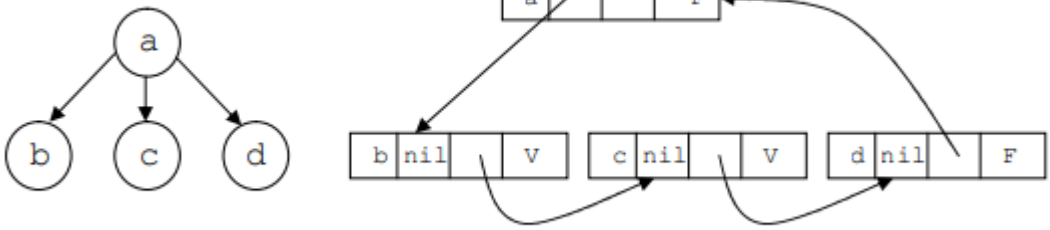


6. Árboles n-arios
7. Escribe un método para la clase árbol (árbol n-ario), que devuelva el número de nodos que forman el árbol.
8. Escribe un método para la clase árbol (árbol n-ario), que calcule el grado del árbol. El grado de un árbol es el máximo de los grados de sus nodos.
9. Escribe un método para la clase árbol (árbol n-ario), que devuelva en forma de lista el resultado del recorrido en anchura del árbol.
10. Escribe una función que, dado un árbol n-ario, devuelva el número de hojas de dicho árbol.
11. Escribe una función booleano que dados dos árboles generales determine si tienen la misma estructura. Por ejemplo, los árboles generales que siguen tienen la misma estructura, aunque, como puede observarse, no coincidan los valores que se almacenan en los nodos.

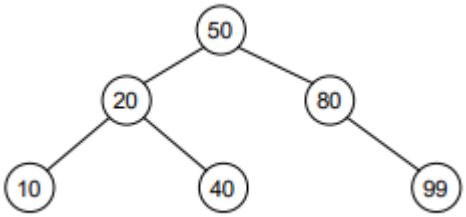




12. Suponemos una implementación del árbol n-ario como la estudiada en clase. Aunque en un principio la referencia al siguiente hermano estaría a nulo en el último hijo, podemos aprovecharla para apuntar al padre. Añadimos un campo extra (booleano) a cada nodo para indicar si existen más hermanos. Construye una clase que únicamente contenga el código del método padre. Dado un elemento, dicho método devuelve, el árbol n-ario que tiene como raíz al padre del nodo que contiene el elemento que se pasa como parámetro. Puedes suponer que el elemento se encuentra en el árbol.



13. .Muestra el resultado de insertar 20, 16, 44, 57, 93, 32, 65, 19, 8 y 17 en un árbol AVLinicialmente vacío.
14. Dibuja el árbol AVL que resulta a partir de la siguiente entrada de datos: 35, 18, 9, 58, 14, 49, 51, 67, 60.
15. Dibuja el árbol AVL que resulta a partir de la siguiente entrada de datos: 24, 14, 6, 35, 59, 17, 21, 32, 4, 7, 15, 22.
16. Dibuja el árbol AVL que resulta de realizar las siguientes inserciones: 13, 7, 21, 15, 27, 18, 4, 11, 30. A continuación elimina los elementos: 13, 4, 15.
17. Partiendo del siguiente árbol AVL, realizar las operaciones que se detallan, marcando para cada nodo su factor de equilibrio en cada momento. En caso de producirse desequilibrio, indicar la causa y explicar con detalle qué operación se ha utilizado para resolverlo. Inserciones: 120, 45, 48 y 30. Eliminaciones: 40, 50 y 80



18. En el árbol AVL siguiente indique la(s) afirmación(es) ciertas:

a) El árbol no está equilibrado en altura.

b) Sólo las hojas tienen factor de equilibrio nulo.

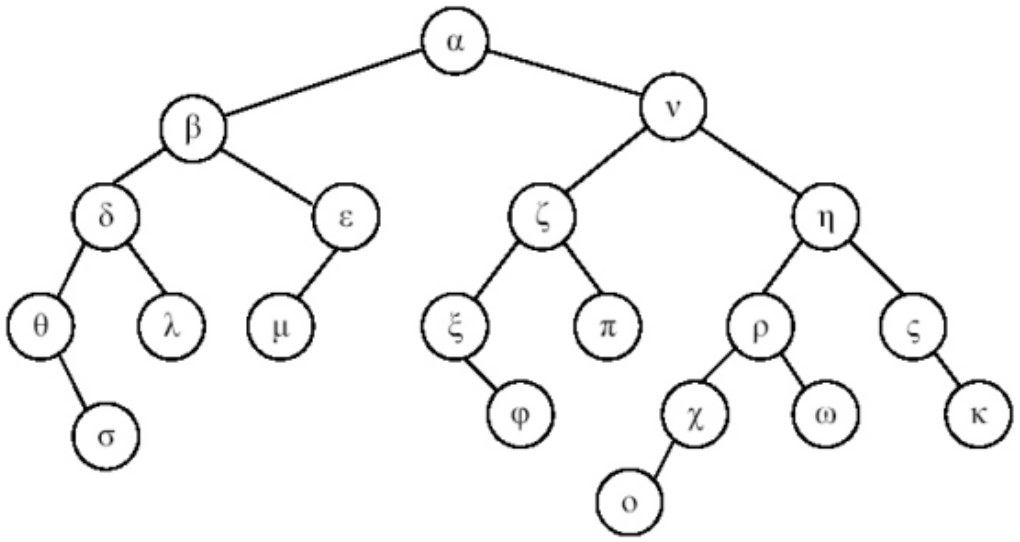
c) Si eliminamos el símbolo v sustituyéndolo por uno de clave

menor, e hijo izquierdo de ζ será nulo.

d) Si eliminamos el símbolo v sustituyéndolo por uno de clave

menor, e hijo izquierdo de ζ será ϕ.

e) Ninguna de las anteriores.



19. Realizar la traza del algoritmo Ordenación rápida o QuickSort (con todos los pasos posibles). Para el siguiente array de números enteros:

32	10	45	39	12	2	8	22	16	15

20. Realizar la traza del algoritmo Ordenación rápida o QuickSort (con todos los pasos posibles). Para el siguiente array de números enteros:

79	21	15	99	88	65	75	85	76	46	84	24

21. Sabiendo que el factor de carga de una tabla Hash es el cociente entre claves almacenadas y tamaño de la tabla. Escribir una función que devuelva el factor de carga de una tabla Hash implementada con dispersión abierta
22. Escribir un procedimiento que dada una lista de listas o lista generalizada (A) de enteros sin ciclos ni sub listas compartidas, cree otra lista generalizada idéntica a la lista A pero sin los átomos negativos.