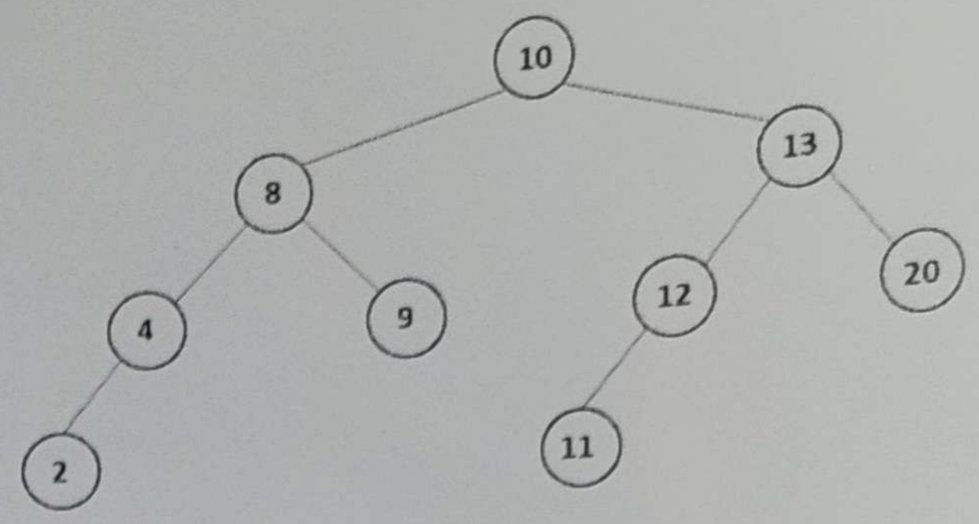
Primer Recuperatorio de Programación 3 - TUDAI

29/6/2022

Ejercicio 1

Dado un árbol binario de búsqueda con valores enteros, como el que se muestra a la derecha, **implementar un algoritmo en Java** que permita verificar si la diferencia entre cada nodo interno del árbol y su sucesor inmediato (**sin considerar las hojas**) no es mayor a un valor K.



Por ejemplo, para un valor de K = 3, el árbol de la imagen NO cumple dicha propiedad, ya que la conexión 8 -> 4 supera a K. En cambio, para el valor K = 5, el árbol de la imagen SI cumple con la propiedad, puesto que no existen dos nodos internos conectados entre sí cuya diferencia sea mayor a 5.

Ejercicio 2

Sudoku: Dado un tablero de 9x9 que internamente está dividido en 9 regiones de 3x3, **donde ya se han colocado algunos números**, se desea determinar qué números del 1 al 9 hay que ubicar en cada casillero vacío de forma de completar el tablero pero sin repetir números por filas, columnas, ni dentro de cada región.
Se solicita plantear un algoritmo mediante estrategia **Backtracking** que resuelva el problema antes mencionado.

a) Realizar el **diagrama del árbol** de exploración de un algoritmo que resuelva el problema mediante la estrategia de Backtracking. Indique en el diagrama qué *información lleva cada estado*, que *decisiones se toma en cada paso* y cuando se llega a un *estado final*.
b) Desarrollar un **pseudo-código** del algoritmo de Backtracking planteado.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Ejercicio 3

Las autoridades de una ciudad deciden construir una red de subterráneos para resolver los constantes problemas de tráfico. La ciudad ya cuenta con N estaciones construidas, pero todavía no tienen ningún túnel que conecte ningún par de estaciones entre sí. La red de subterráneos que se construya debe incluir a **todas las estaciones** (es decir, que de cualquier estación pueda llegar a cualquier otra estación, **ya sea de manera directa o atravesando otras estaciones**). Sin embargo, debido al acotado presupuesto, las autoridades desean **construir la menor cantidad de metros de túnel posibles**. Se tiene como información cuántos metros de túnel serían necesarios para conectar de manera directa cada par de estaciones existentes.

Plantee en **pseudo-código** un algoritmo que mediante una estrategia **Greedy** proponga la lista de túneles a construir para conectar todas las estaciones. **Nota: Considere que los túneles que se construyen son de ida y vuelta.**
Luego, considerando el algoritmo Greedy planteado, explique con sus palabras:

a) ¿Cuáles son los *candidatos* del algoritmo?
b) ¿Cuál es la *estrategia de selección greedy* de candidatos?
c) ¿Cuándo es *factible* agregar un candidato a la solución?

Ejercicio 4

Dada la siguiente estructura de hashing separado con crecimiento lineal, con M=5, rp=2, rs=1, ρ_d=0,8. La frontera se encuentra en f=1 y la función de hashing utilizada es h(x) = x mod M.

	26	7		24	
10	11	12	23	4	5
0	1	2	3	4	5

f=1

Insertar las claves < 15, 28, 21 > en ese orden.
Luego de cada inserción explique qué operaciones realizó. Muestre el ρ_t y la estructura resultante.

Ejercicio 5

Diga para cada una de las siguientes afirmaciones si son **verdadera o falsa, y justifique su respuesta en cada caso**:

a) Si se tuviera una función de hashing perfecto sobre una estructura de hashing cerrado con M baldes y rp=2, se estaría desperdiciando espacio en dicha estructura.
b) El algoritmo de Dijkstra si es aplicado a un grafo dirigido no conexo, siempre resultará en un arreglo de distancias con al menos un valor infinito.
c) El árbol de ejemplo del Ejercicio 1 no es un árbol binario balanceado.