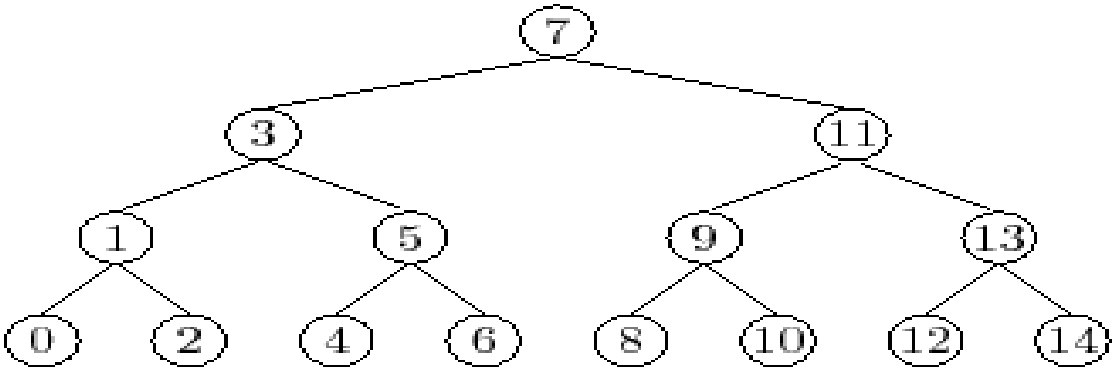
Nama: Samuel Lee

1. Explain the differences between linear and non-linear data structures!
   Perbedaanya terletak di urutan indexnya, untuk linear memiliki index yang berurutan seperti array. Contohnya: linkedlist
   Untuk non-linear urutan indexnya tidak berurutan apabila digambarkan seperti gambar tree.

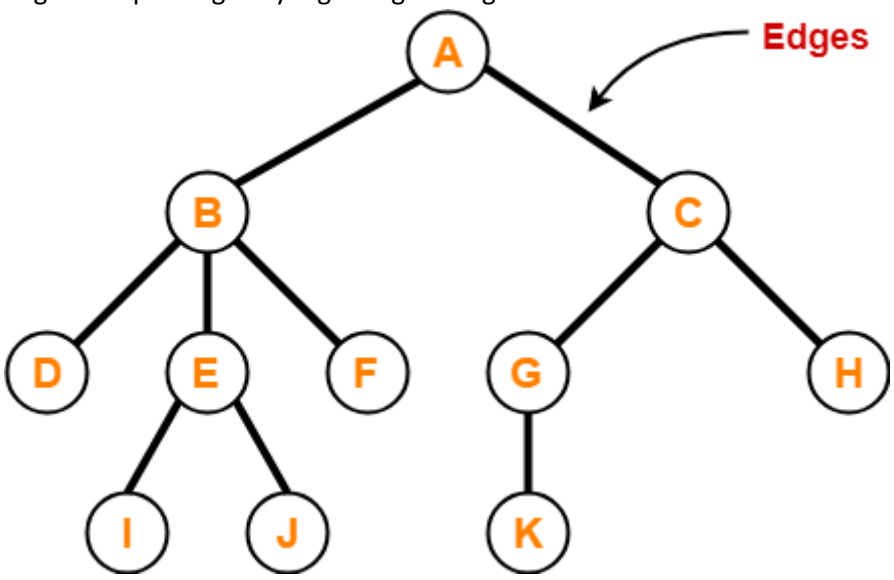
2. Describe the following terminology in a tree: base root, key, edge, sibling, parent, child, and leaf!
   Base root artinya Node yang paling pertama atau paling atas pas kita pertama kali membuat looping atau print dari nodenya. Gambar



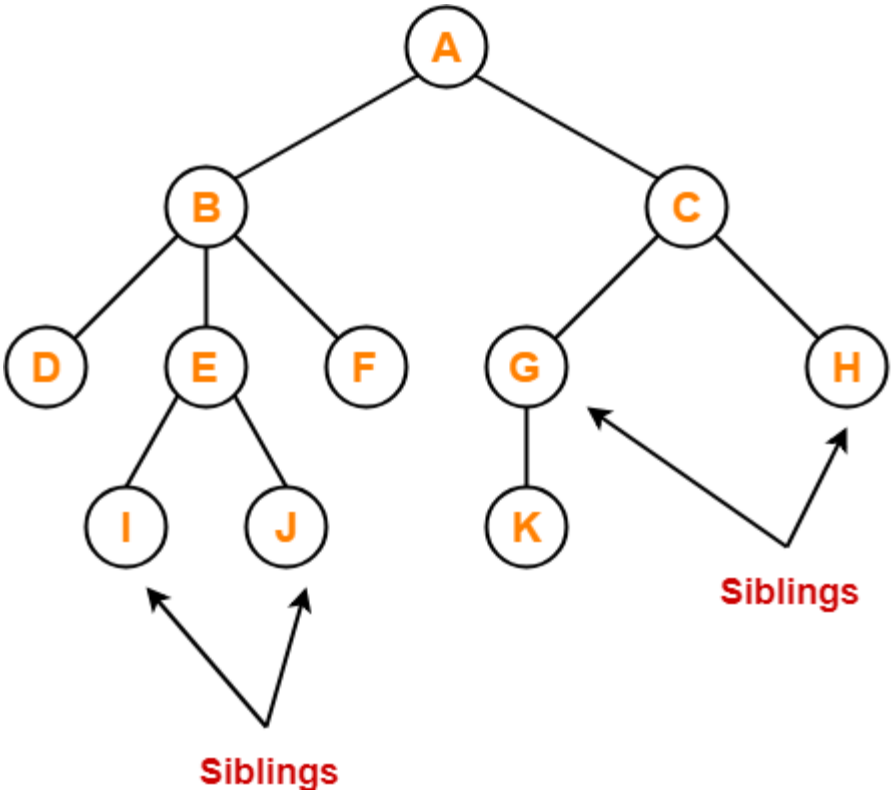
   Angka 7 merupakan base root.

   Key dapat diakatakan sebagai atribut pembagi dalam suatu tree jadi misal, pada gambar diatas Node 7 membagi treenya menjadi 2 arah. Sebelah kiri untuk yang nilainya lebih kecil dari 7 dan sebelah kana untuk yang nilainya lebih besar dari 7.

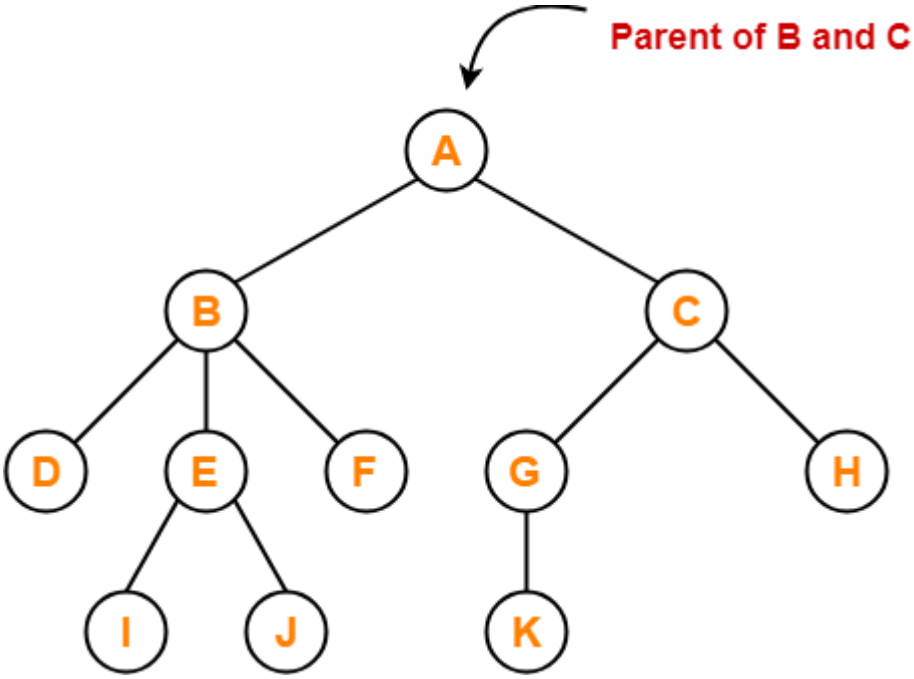
   Edge merupakan garis yang menghubungkan antara 2 Node



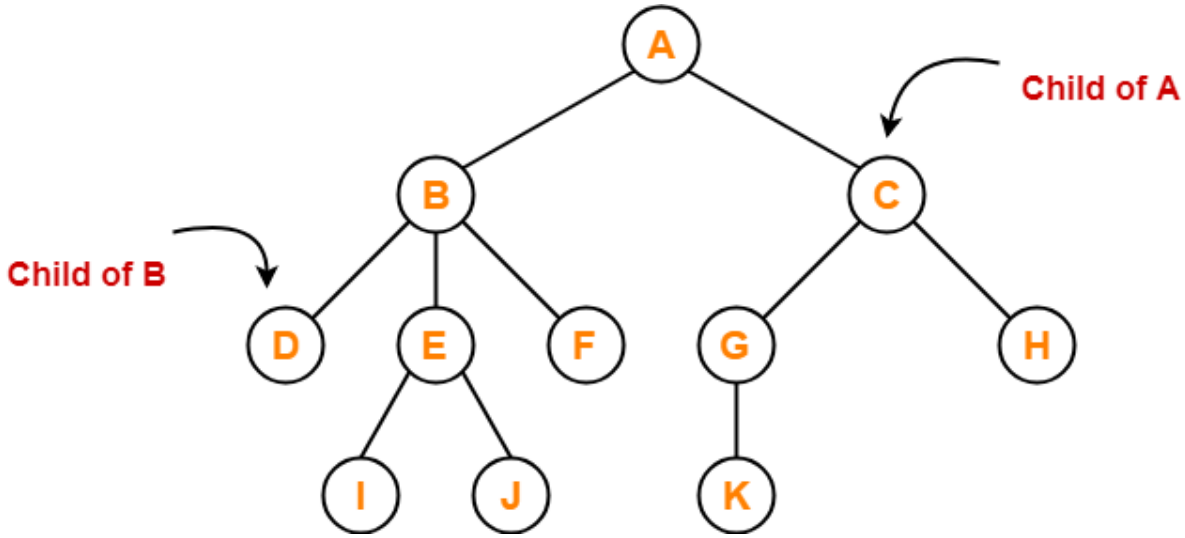
   Sibling merupakan Node yang memiliki orang tua yang sama sebagai Node, contoh:



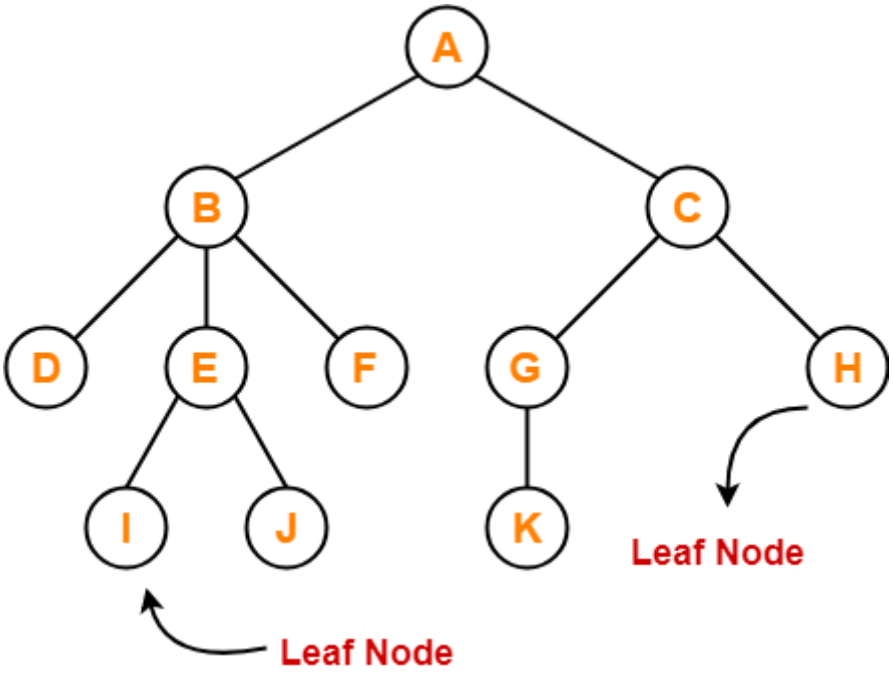
Parent merupakan Node yang terhubung dengan node lain yang berada 1 level dibawahnya, contoh:



Child merupakan Node yang merupakan turunan dari beberapa Node, semua node kecuali base root merupakan Node anak.



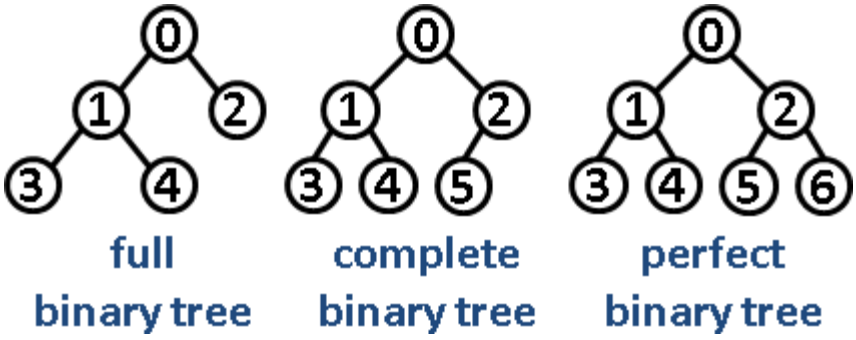
Leaf merupakan Node yang tidak memiliki anak dibawahanya, contoh:



3. Explain the following types of binary trees: full, complete, and prefect!
   Full type binary tree merupakan binary tree yang setiap nodenya memiliki 0 dan 2 anak.

   Sebuah binary tree dikatakan sebagai complete apabila semua level dari tree tersebut terisi penuh kecuali level terakhir boleh tidak terisi.

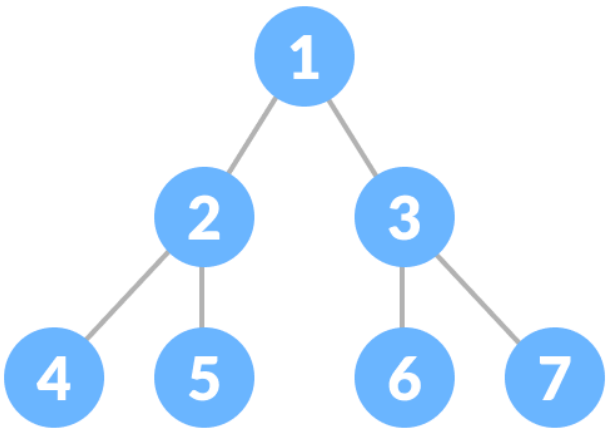
   Binary tree dikatakan perfect binary tree apabila semua nodenya memiliki 2 anak dan semua leaf berada pada level yang sama



4. What makes a tree balanced?

   Sebuah tree bisa menjadi balanced kalau perbedaan height antara left sama right subtreenya <= 1.

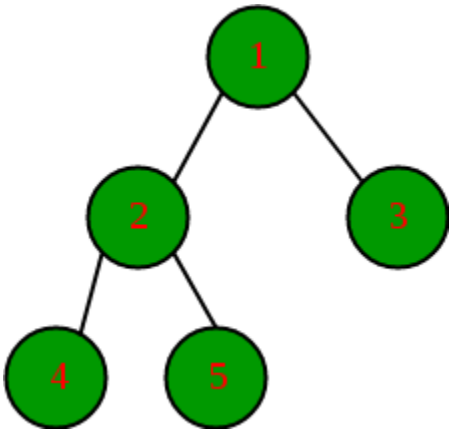
5. Explain the four properties of a binary tree!
   - Number of Nodes
   a. Max nodes at level k
      Menggunakan perfedtc binary tree karena kita mencari maksimal node pada level k



      Level 0: ada 1 node
      Level 1: ada 2 node
      Level 2: ada 4 node
      Jadi hubungan antara level k dan jumlah node apabila dirumuskan menjadi 2^K
   b. Max nodes in a tree with level k
      Artinya kita ingin mencar jumlah node maksimal dalam sebuah tree
      Jadi apabila kita menggunakan gambar diatas sebagai contoh, maka rumus untuk mencari jumlah maksimal node dalam sebuah tree adalah:
      Max = $2^{k+1} – 1$
      Jadi, apabila dihitung menjadi
      Max = $2^{3+1} – 1 = 8 – 1 = 7$

- Height of tree

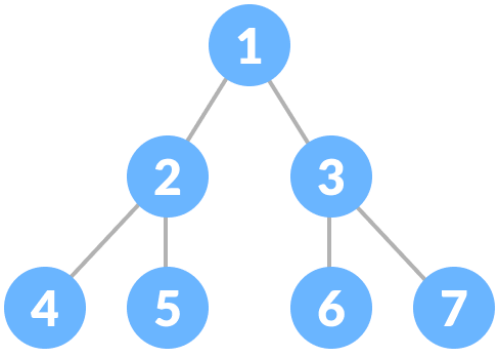
c. Min level untuk n nodes
Bertujuan untuk membuat sebuah tree menjadi sekompleks mungkin tanpa melanggar aturan binary tree
Jadi missal diberi kan 5 node 1,2,3,4,5



Tree diatas memiliki 5 node dan kita ingin membuat tree tersbut menjadi 2 level, maka cara yang perlu digunakan adalah kita menggunakan log2(n)
Jadi kita perlu menghitung 2 pangkat berapa yang menghasilkan angka 5.

d. Max level untuk n nodes
Untuk mencari max level maka kita akan menggunakan linkedlist sehinggan tree tersebut akan berurutan kebawah yang dimulai dari
1 level 0
2 level 1
3 level 2
4 level 3
5 level 4
Sehingga levelnya sekarang menjadi 4 level diperolej dari hasil n-1.

6. Explain the intuition of implementing a binary tree using an array!
Missal kita memiliki sebuah tree

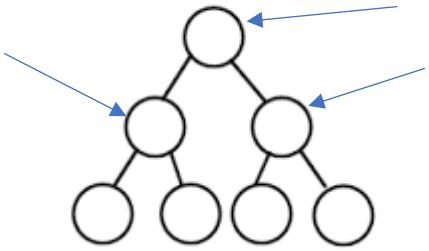


Untuk pergi ke left maka kita akan menggunakan rumus 2p+1, sedangkan untuk kearah right maka rumusnya menjadi 2p+2.
Dan sebuah array

0	1	2	3	4	5	6
1	2	3	4	5	6	7

Jadi misalkan index adalah p dan kita ingin menerapkan binary tree ke dalam array
Maka rumus left menjadi: 2(index)+1
Rumus right: 2(index)+2
Contoh:
Apabila berada di index ke 0
Untuk left akan menjadi: 2(0)+1 = 1
Untuk right akan menjadi: 2(0)+2 = 2
Jadi left pada binary tree akan menempati array dengan index 1 pada array yang bernilai 2 dan right akan menempati array dengan index 2 yang bernilai 3.

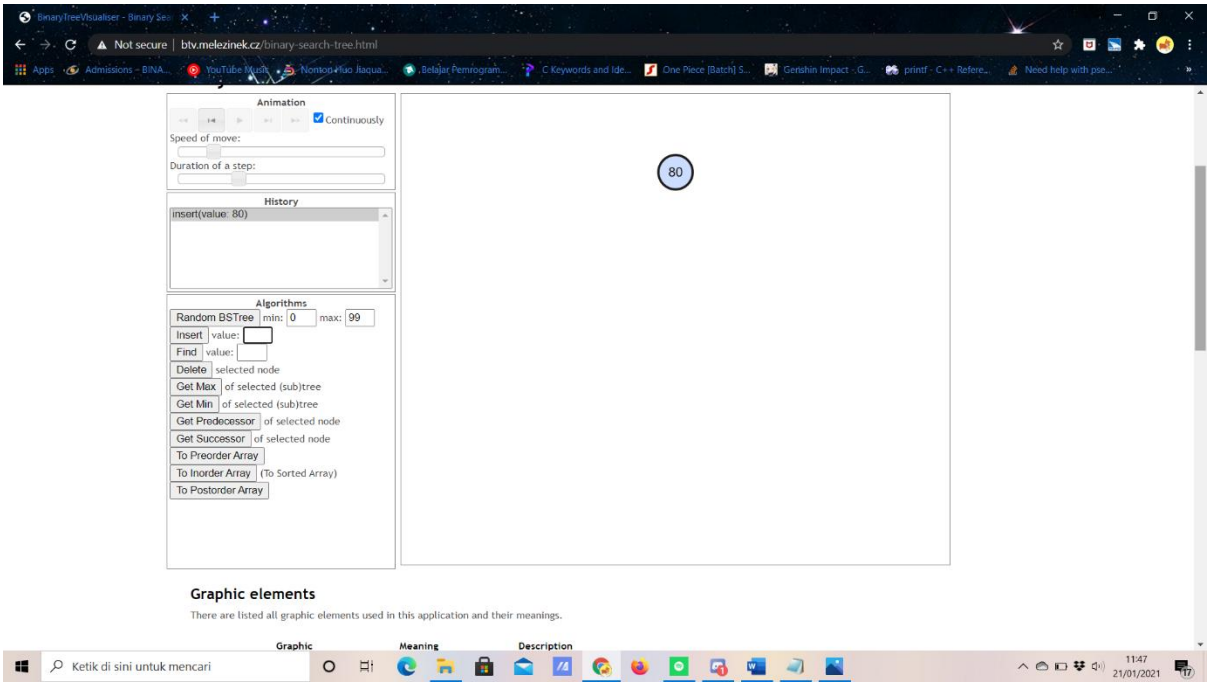
7. Explain the differences between inorder successor and inorder predecessor!



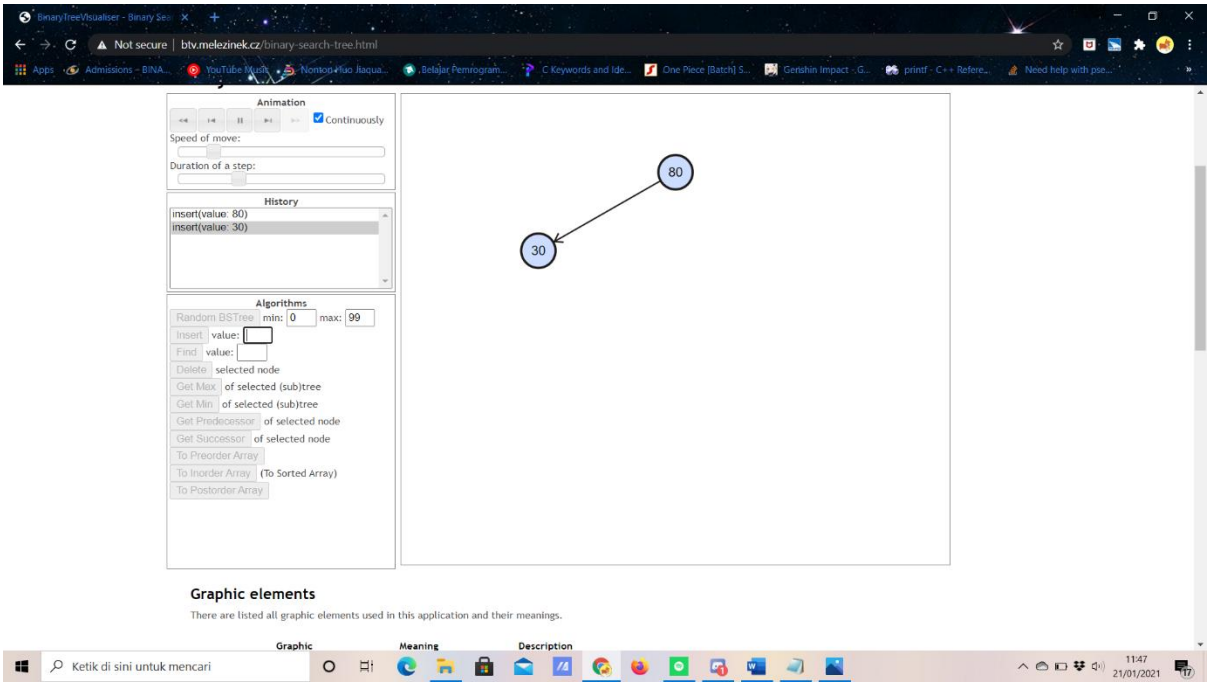
Inorder successor adalah satu node sebelumnya dari root dan inorder predecessor adalah node setelah root atau penerus dari root
jadi kalau diurutkan menjadi 1,2,3,4,5,6,7 misal root dari tree ini adalah 4, maka inorder successornya adalah 3 dan inorder predecessornya adalah 5.

Kegunaan dari menentukan hal tersebut adalah Ketika kita ingin mendelete node yang memiliki 2 anak.

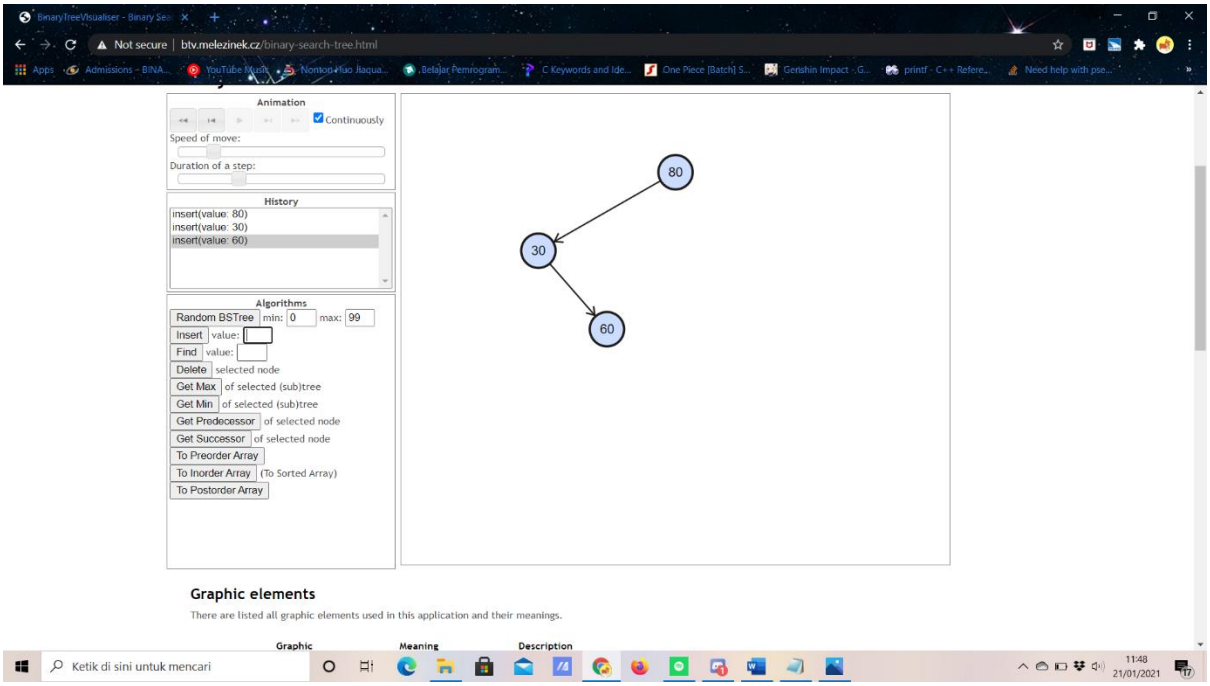
8. Draw the following binary search tree step by step (14 pictures):
   - Insert 80, 30, 60, 50, 75
   - Delete 60, 30, 75
   - Insert 65, 30, 35
   - Delete 80, 65, 35



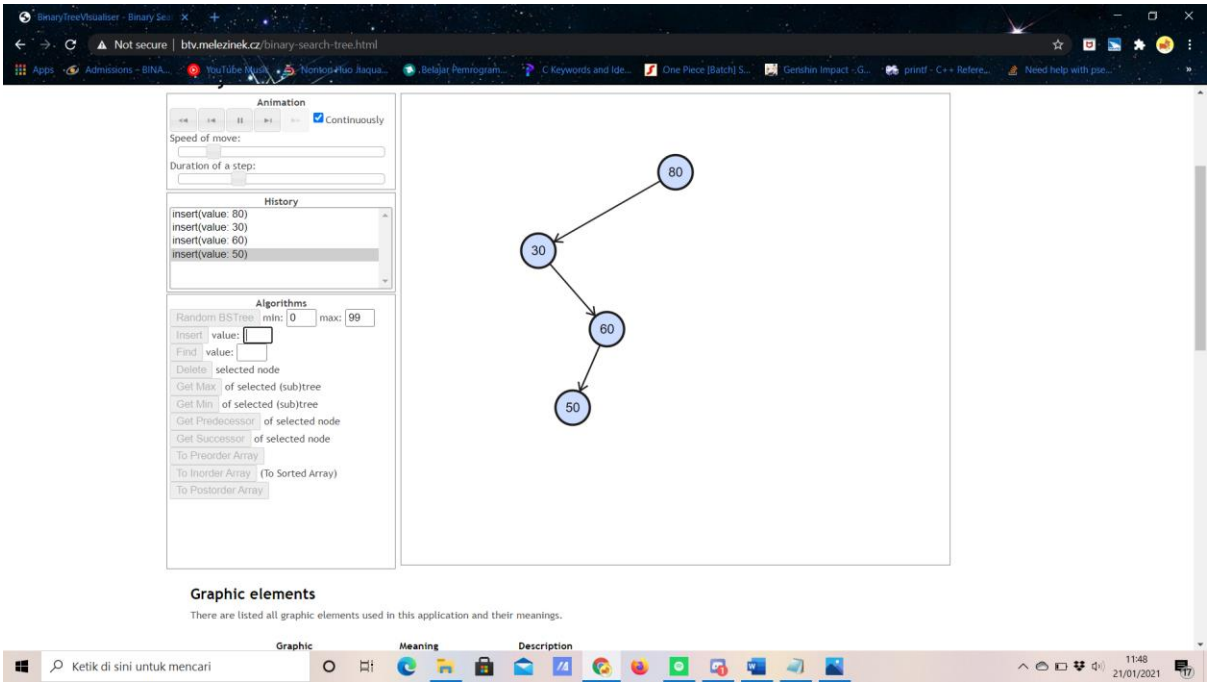
Balance Factor = 0



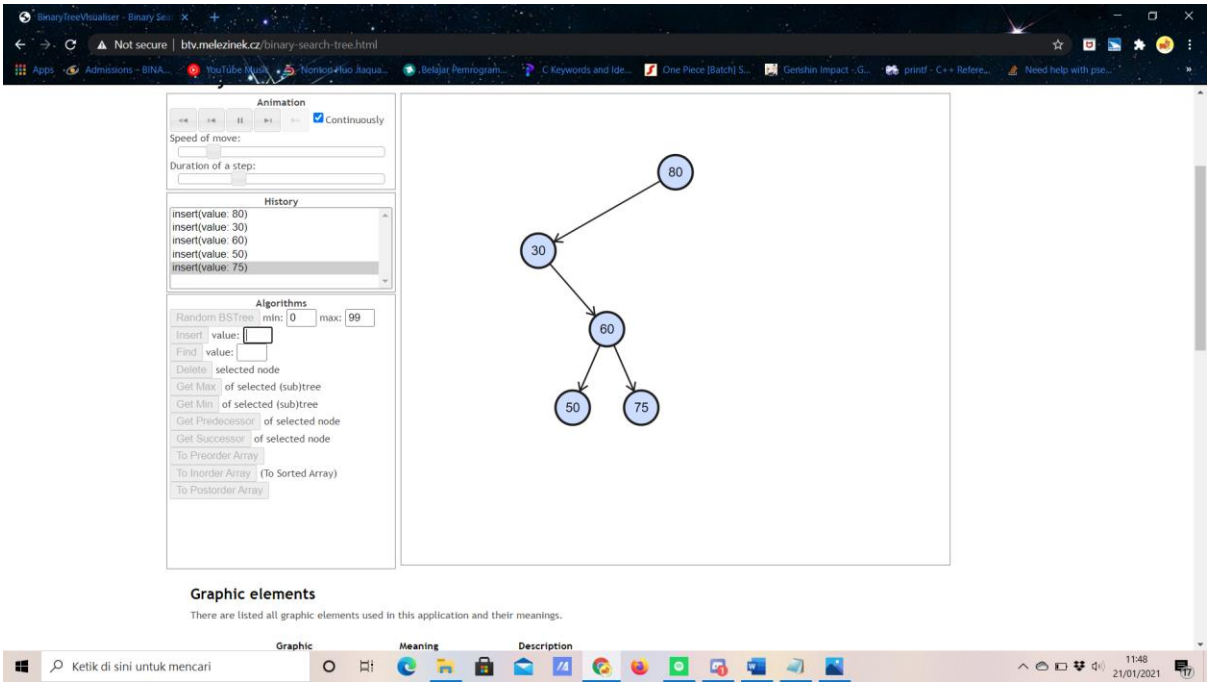
Balance Factor = 0



Balance Factor = 1

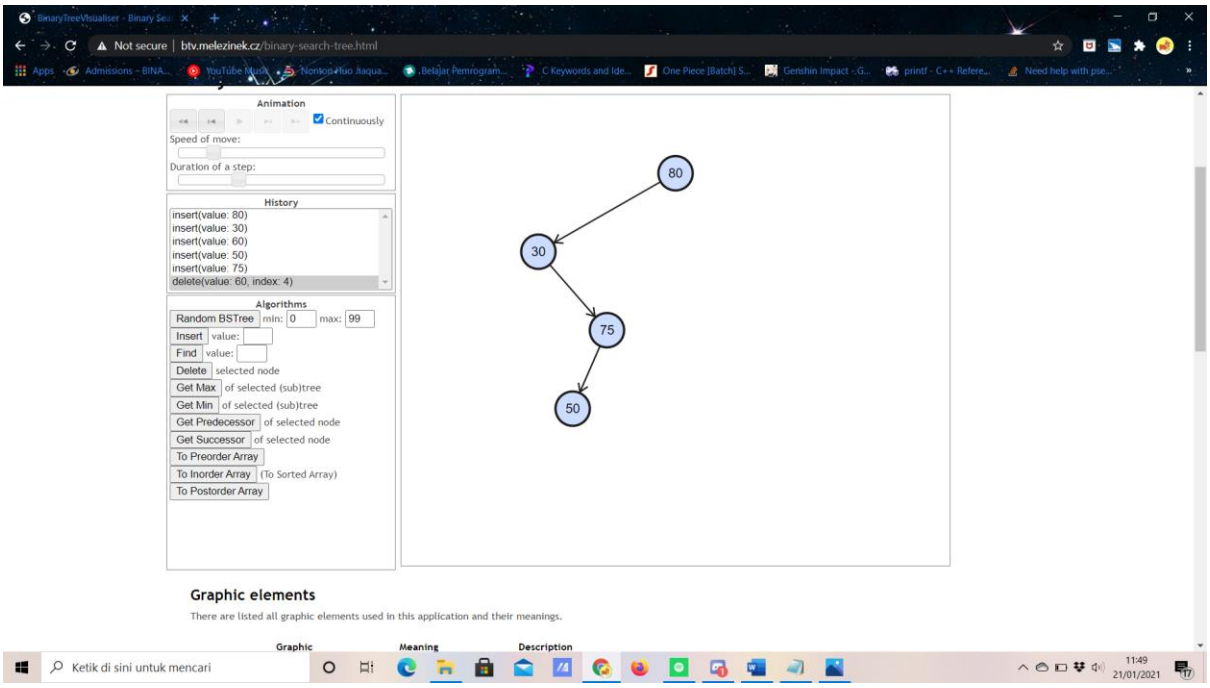


Balance Factor = 2

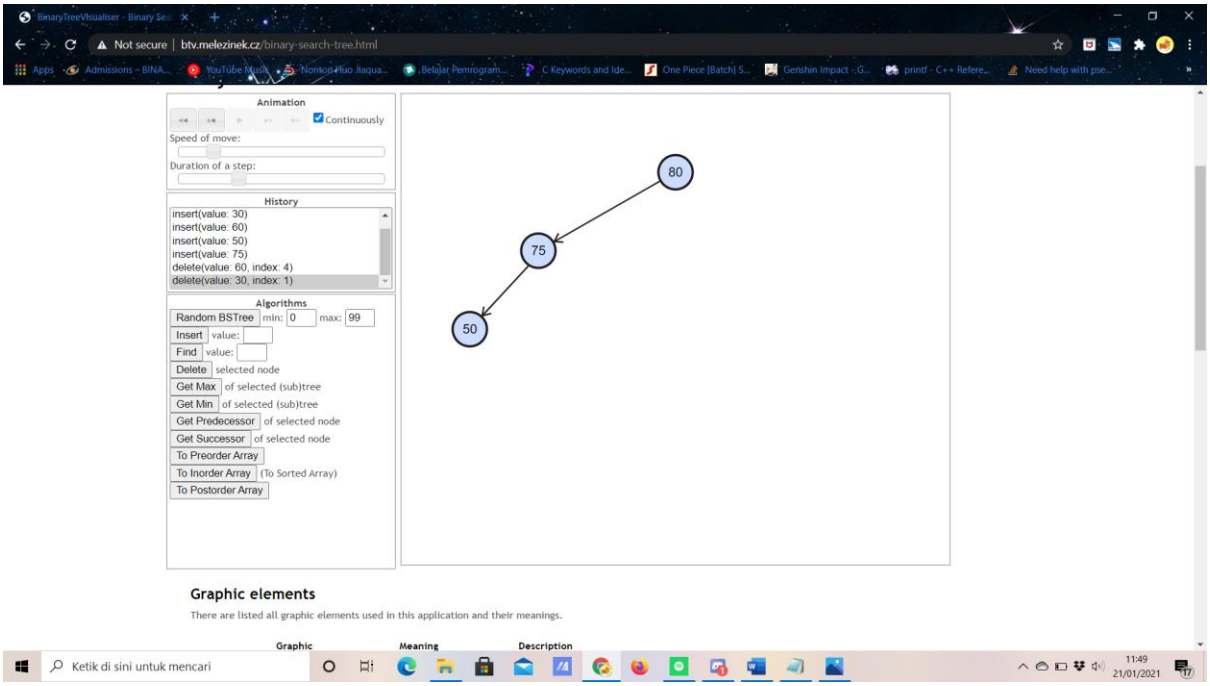


Balance Factor = 0

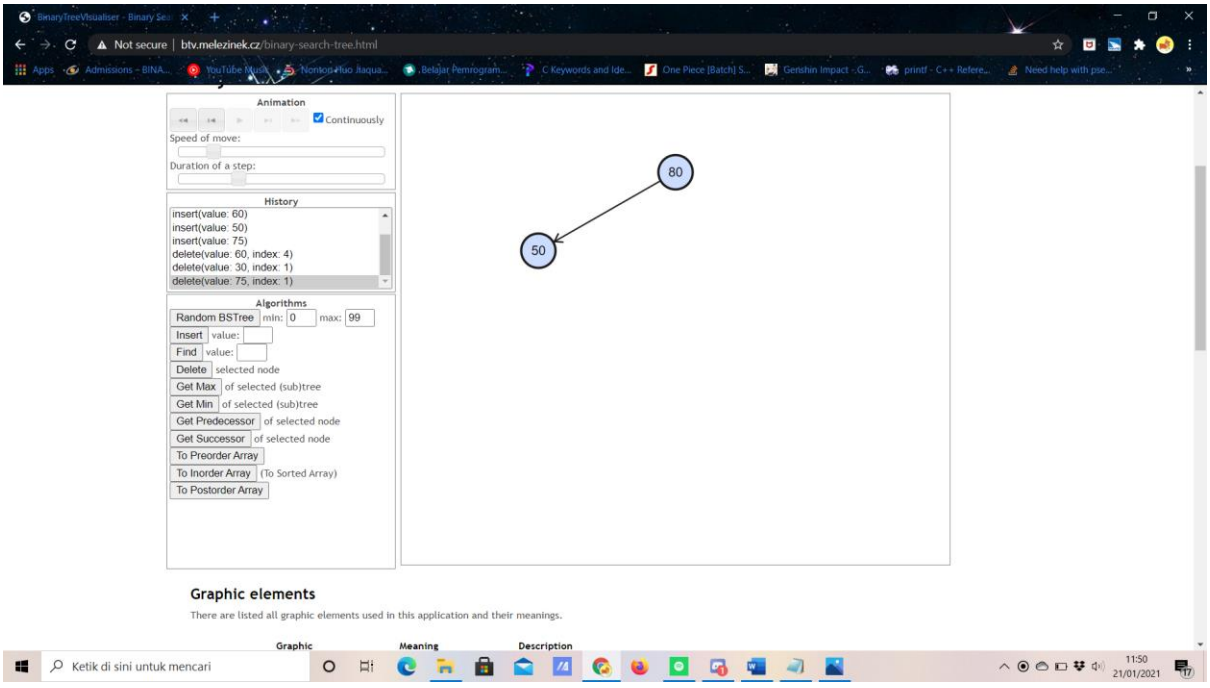
Delete 60, 30, 75



Balance Factor = 2

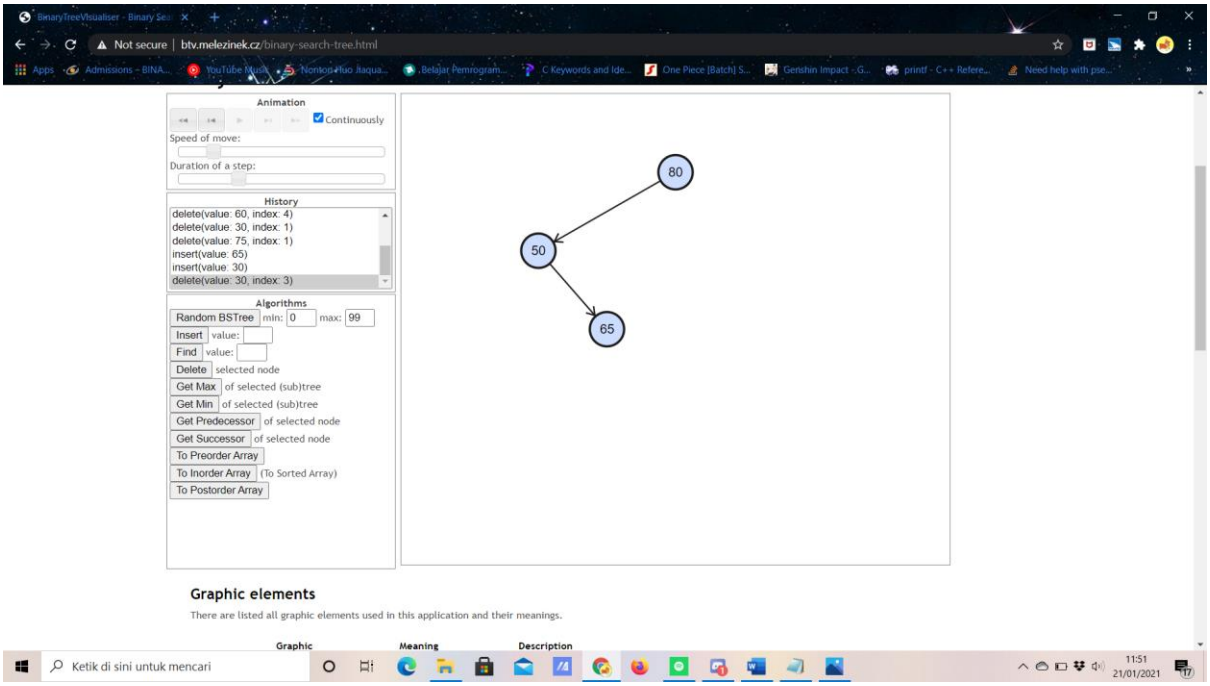


Balance Factor = 1

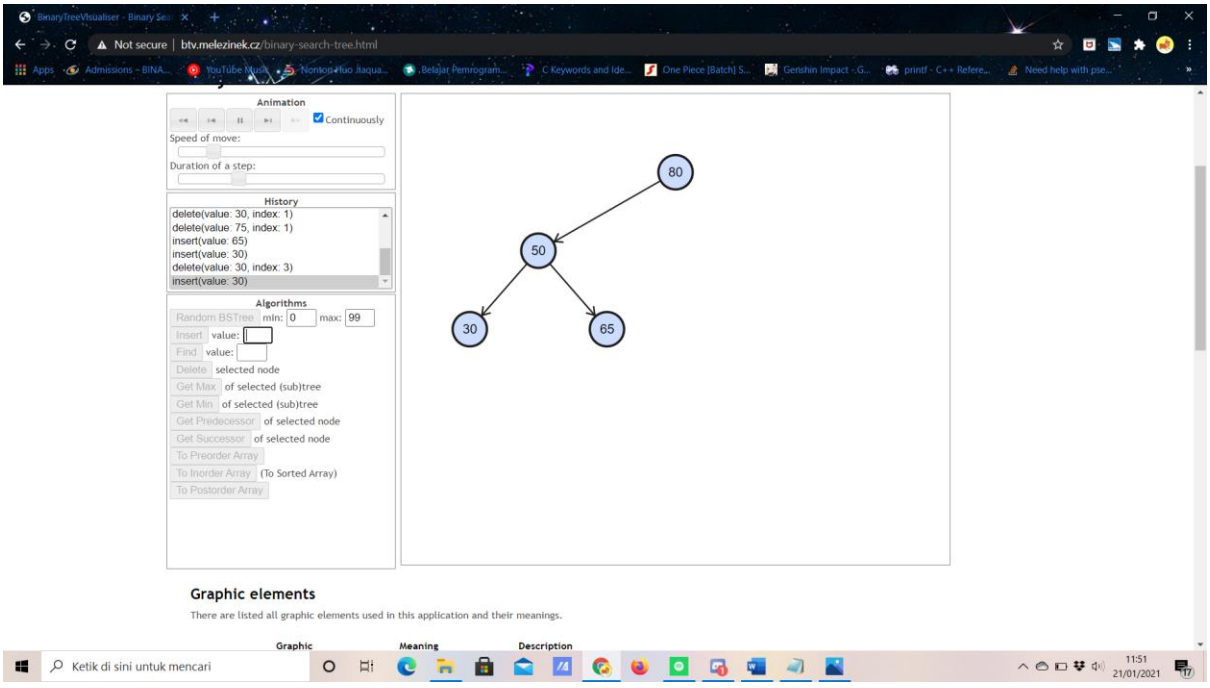


Balance Factor = 0

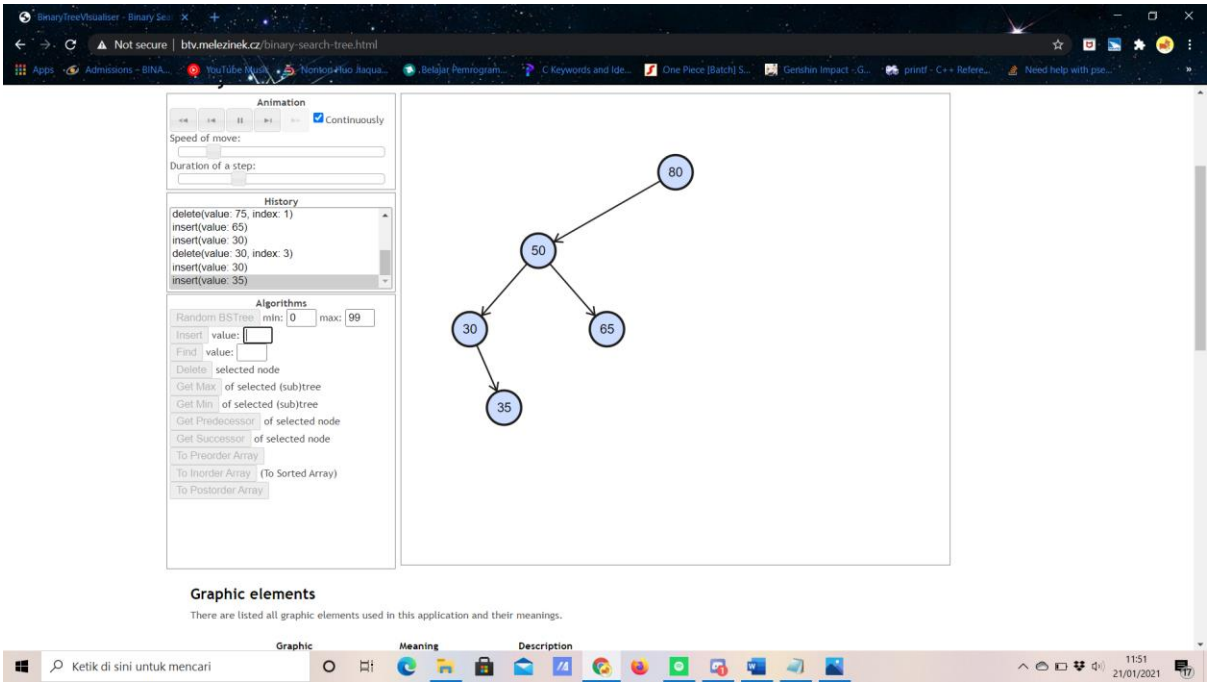
Insert 65, 30, 35



Balance Factor =1

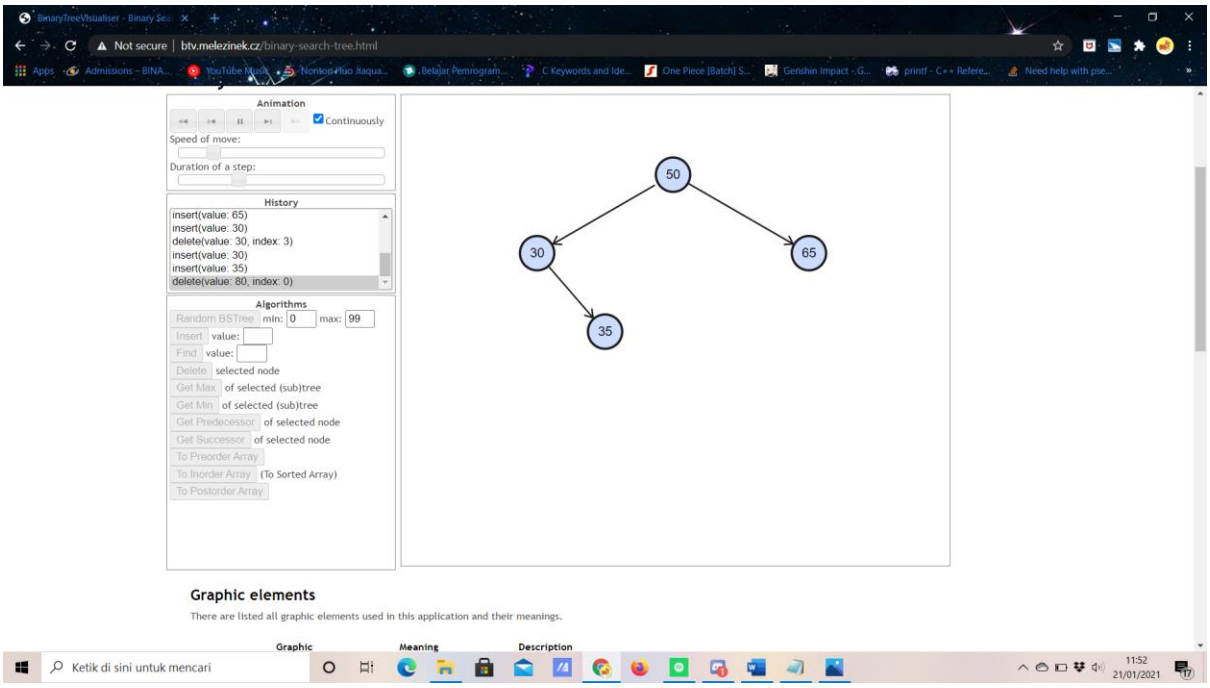


Balance Factor = 1

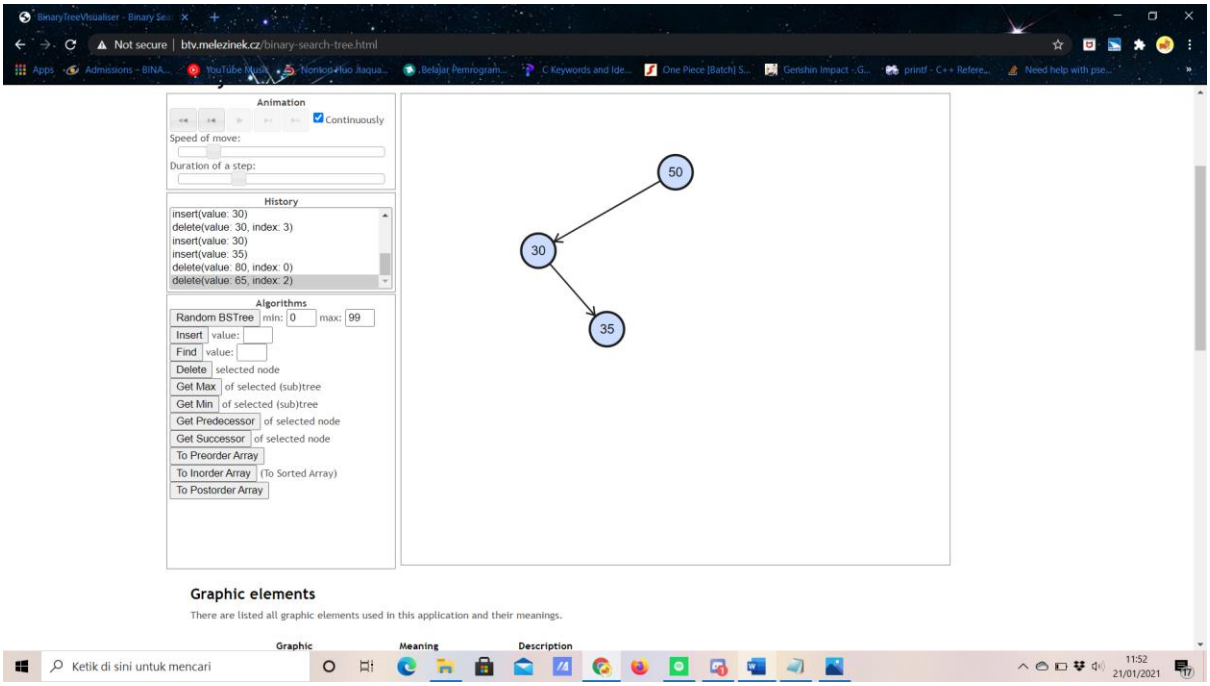


Balance Factor = 1

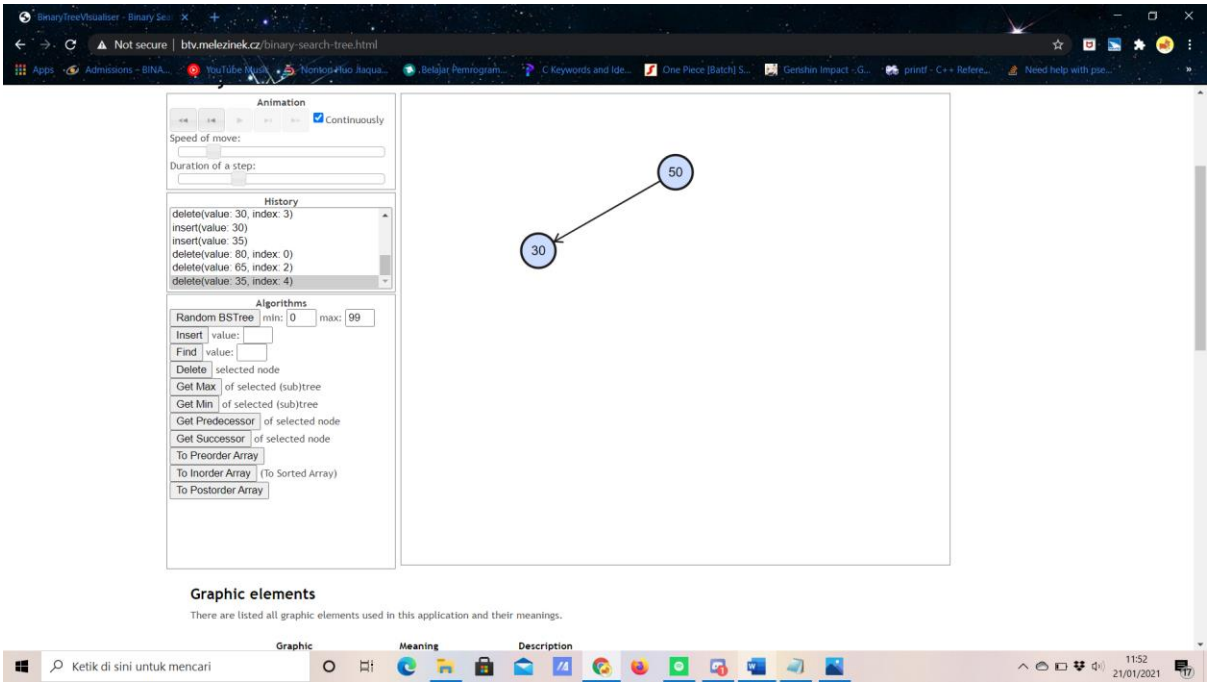
Delete 80, 65, 35



Balance Factor = 1



Balance Factor = 1



Balance Factor = 0