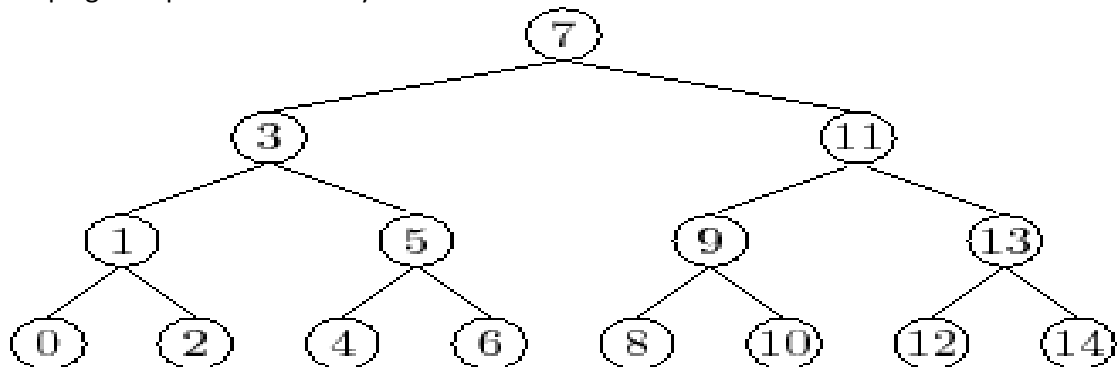


Nama: Samuel Lee

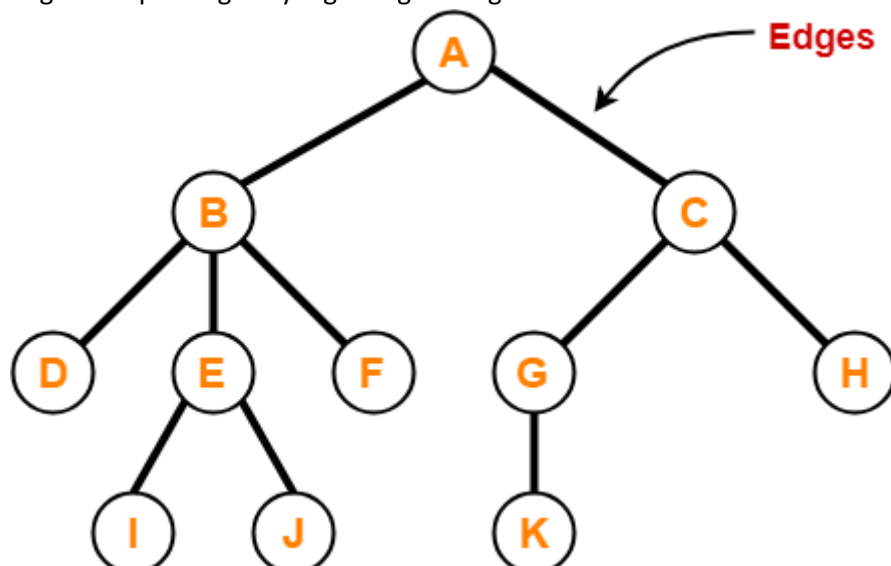
1. Explain the differences between linear and non-linear data structures!  
Perbedaannya terletak di urutan indexnya, untuk linear memiliki index yang berurutan seperti array. Contohnya: linkedlist  
Untuk non-linear urutan indexnya tidak berurutan apabila digambarkan seperti gambar tree.
2. Describe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



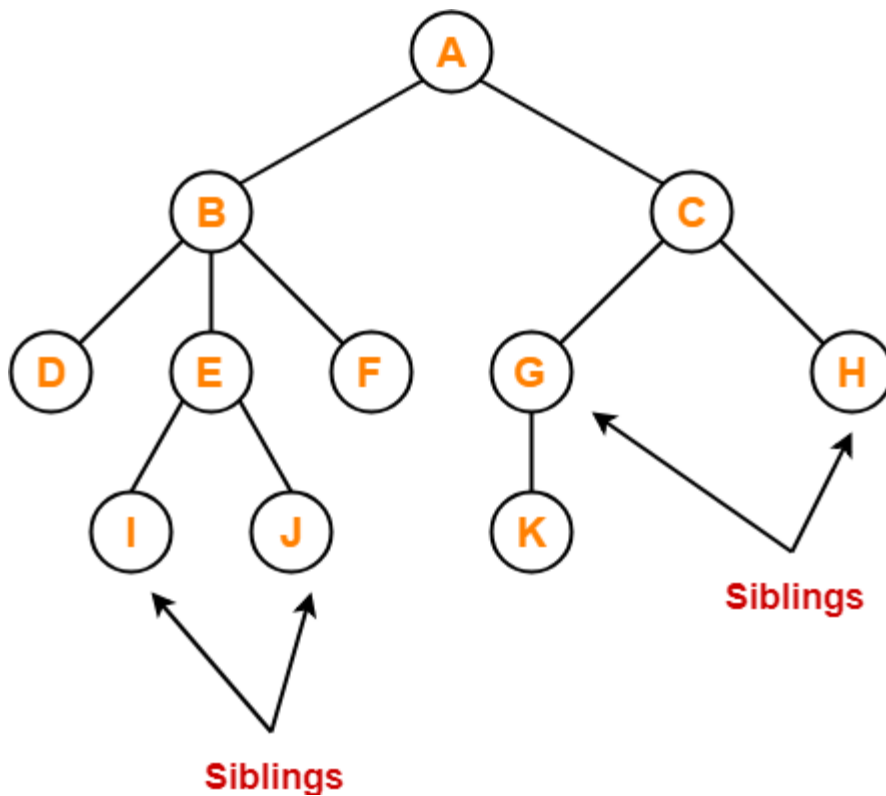
Angka 7 merupakan base root.

Key dapat dikatakan 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 kanan untuk yang nilainya lebih besar dari 7.

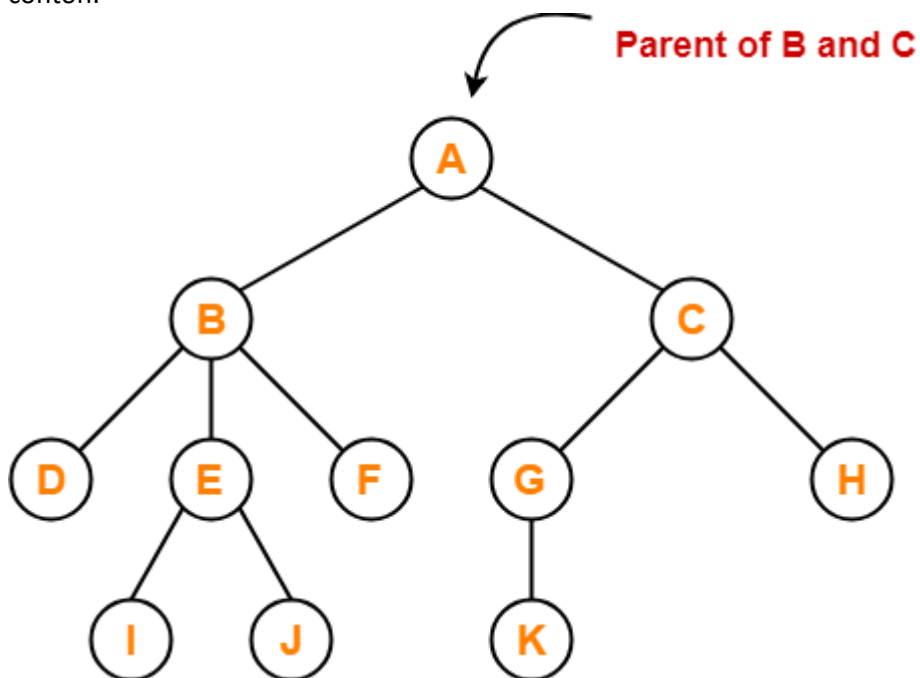
Edge merupakan garis yang menghubungkan antara 2 Node



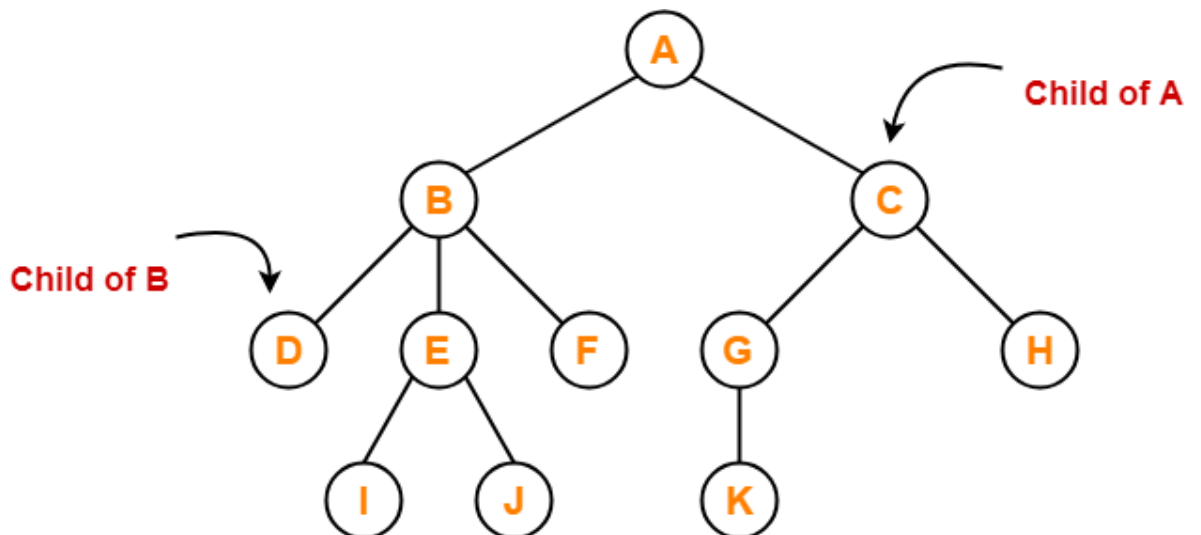
Sibling merupakan Node yang memiliki orang tua yang sama sebagai Node, contoh:



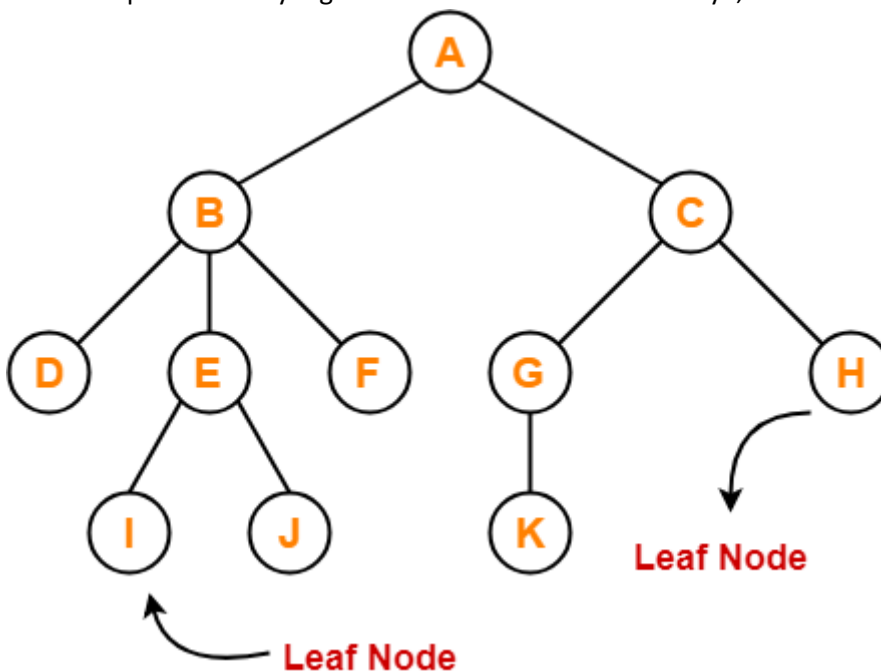
Parent merupakan Node yang terhubung dengan node lain yang berada 1 level dibawahnya, contoh:



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



Leaf merupakan Node yang tidak memiliki anak dibawahnya, contoh:

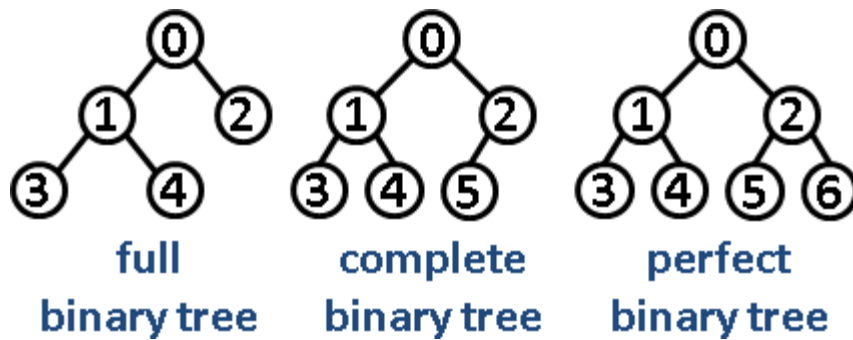


3. Explain the following types of binary trees: full, complete, and perfect!

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?

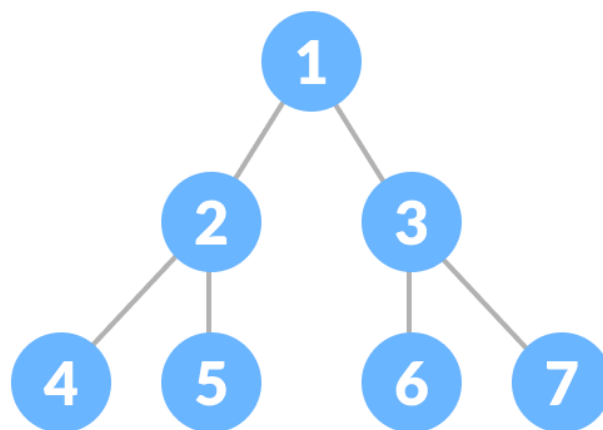
Sebuah tree bisa menjadi balanced kalau perbedaan height antara left sama right subtreenya  $\leq 1$ .

5. Explain the four properties of a binary tree!

- Number of Nodes

a. Max nodes at level k

Menggunakan perfect 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 mencari 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:

$$\text{Max} = 2^{k+1} - 1$$

Jadi, apabila dihitung menjadi

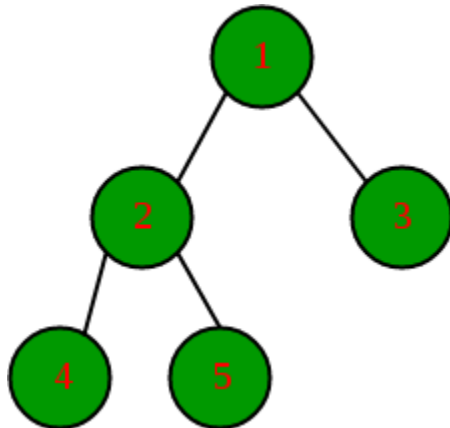
$$\text{Max} = 2^{3+1} - 1 = 8 - 1 = 7$$

- Height of tree

c. Min level untuk n nodes

Bertujuan untuk membuat sebuah tree menjadi sekompleks mungkin tanpa melanggar aturan binary tree

Jadi misal diberi kan 5 node 1,2,3,4,5



Tree diatas memiliki 5 node dan kita ingin membuat tree tersebut menjadi 2 level, maka cara yang perlu digunakan adalah kita menggunakan  $\log_2(n)$

Jadi kita perlu menghitung 2 pangkat berapa yang menghasilkan angka 5.

d. Max level untuk n nodes

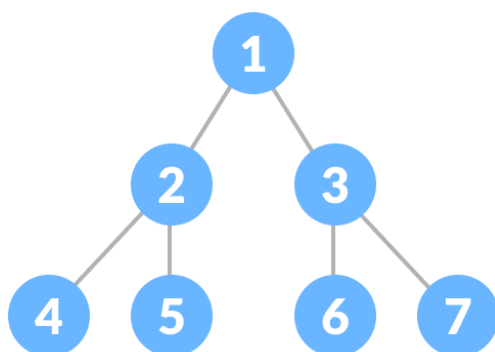
Untuk mencari max level maka kita akan menggunakan linkedlist sehingga tree tersebut akan berurutan kebawah yang dimulai dari

- 1 level 0
- 2 level 1
- 3 level 2
- 4 level 3
- 5 level 4

Sehingga levelnya sekarang menjadi 4 level diperoleh dari hasil  $n-1$ .

6. Explain the intuition of implementing a binary tree using an array!

Misal 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:  $2(\text{index})+1$

Rumus right:  $2(\text{index})+2$

Contoh:

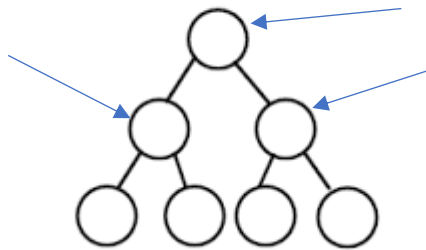
Apabila berada di index ke 0

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 5 dan inorder predecessornya adalah 3.

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

The screenshot shows the BinaryTreeVisualizer application interface. The main canvas displays a single node with the value 80. The left sidebar contains several sections:
 

- Animation:** Includes controls for 'Speed of move' and 'Duration of a step', with a 'Continuously' checkbox checked.
- History:** A list showing 'Insert(value: 80)'.
- Algorithms:** A list of operations including 'Random BSTree', 'Insert', 'Find', 'Delete', 'Get Max', 'Get Min', 'Get Predecessor', 'Get Successor', 'To Preorder Array', 'To Inorder Array', and 'To Postorder Array'. Each operation has a corresponding input field or selection mechanism.

**Graphic elements**  
 There are listed all graphic elements used in this application and their meanings.

Graphic	Meaning	Description

Balance Factor = 0

The screenshot shows the BinaryTreeVisualizer application interface. The main canvas displays a tree structure with a root node (80) and a left child node (30). The left sidebar contains several sections:
 

- Animation:** Includes controls for 'Speed of move' and 'Duration of a step', with a 'Continuously' checkbox checked.
- History:** A list showing 'Insert(value: 80)' and 'Insert(value: 30)'.
- Algorithms:** A list of operations including 'Random BSTree', 'Insert', 'Find', 'Delete', 'Get Max', 'Get Min', 'Get Predecessor', 'Get Successor', 'To Preorder Array', 'To Inorder Array', and 'To Postorder Array'.

**Graphic elements**  
 There are listed all graphic elements used in this application and their meanings.

Graphic	Meaning	Description

Balance Factor = 0

BinaryTreeVisualizer - Binary Search Tree

Not secure | btv.melezinek.cz/binary-search-tree.html

Animation

Speed of move:

Duration of a step:

Continuously ☒

History

insert(value: 80)  
insert(value: 30)  
insert(value: 60)

Algorithms

Random BSTree min: 0 max: 99

Insert value:

Find value:

Delete selected node

Get Max of selected (sub)tree

Get Min of selected (sub)tree

Get Predecessor of selected node

Get Successor of selected node

To Preorder Array

To Inorder Array (To Sorted Array)

To Postorder Array

```

graph TD
    80((80)) --> 30((30))
    80 --> 60((60))
  
```

Graphic elements

There are listed all graphic elements used in this application and their meanings.

Graphic Meaning Description

Ketik di sini untuk mencari

11:43 21/01/2021

Balance Factor = 1

BinaryTreeVisualizer - Binary Search Tree

Not secure | btv.melezinek.cz/binary-search-tree.html

Animation

Speed of move:

Duration of a step:

Continuously ☒

History

insert(value: 80)  
insert(value: 30)  
insert(value: 60)  
insert(value: 50)

Algorithms

Random BSTree min: 0 max: 99

Insert value:

Find value:

Delete selected node

Get Max of selected (sub)tree

Get Min of selected (sub)tree

Get Predecessor of selected node

Get Successor of selected node

To Preorder Array

To Inorder Array (To Sorted Array)

To Postorder Array

```

graph TD
    80((80)) --> 30((30))
    80 --> 60((60))
    30 --> 50((50))
  
```

Graphic elements

There are listed all graphic elements used in this application and their meanings.

Graphic Meaning Description

Ketik di sini untuk mencari

11:43 21/01/2021

Balance Factor = 2



BinaryTreeVisualiser - Binary Search Tree

Not secure | btv.melezinek.cz/binary-search-tree.html

Animation

Speed of move:

Duration of a step:

Continuously ☒

History

insert(value: 80)  
insert(value: 30)  
insert(value: 60)  
insert(value: 50)  
insert(value: 75)

Algorithms

Random BSTree min: 0 max: 99

Insert value:

Find value:

Delete selected node

Get Max of selected (sub)tree

Get Min of selected (sub)tree

Get Predecessor of selected node

Get Successor of selected node

To Preorder Array

To Inorder Array (To Sorted Array)

To Postorder Array

Graphic elements

There are listed all graphic elements used in this application and their meanings.

Graphic	Meaning	Description
	Root node	The root of the tree
	Internal node	A node that has at least one child
	Leaf node	A node that has no children
	Edge	A line connecting two nodes

Balance Factor = 0

Delete 60, 30, 75

BinaryTreeVisualiser - Binary Search Tree

Not secure | btv.melezinek.cz/binary-search-tree.html

Animation

Speed of move:

Duration of a step:

Continuously ☒

History

insert(value: 80)  
insert(value: 30)  
insert(value: 60)  
insert(value: 50)  
insert(value: 75)  
delete(value: 60, index: 4)

Algorithms

Random BSTree min: 0 max: 99

Insert value:

Find value:

Delete selected node

Get Max of selected (sub)tree

Get Min of selected (sub)tree

Get Predecessor of selected node

Get Successor of selected node

To Preorder Array

To Inorder Array (To Sorted Array)

To Postorder Array

Graphic elements

There are listed all graphic elements used in this application and their meanings.

Graphic	Meaning	Description
	Root node	The root of the tree
	Internal node	A node that has at least one child
	Leaf node	A node that has no children
	Edge	A line connecting two nodes

Balance Factor = 2

BinaryTreeVisualizer - Binary Search Tree

Not secure | btv.melezinek.cz/binary-search-tree.html

Animation

Speed of move:

Duration of a step:

Continuously ☒

History

```
insert(value: 30)
insert(value: 60)
insert(value: 50)
insert(value: 75)
delete(value: 60, index: 4)
delete(value: 30, index: 1)
```

Algorithms

Random BSTree min:  max:

Insert value:

Find value:

Delete selected node

Get Max of selected (sub)tree

Get Min of selected (sub)tree

Get Predecessor of selected node

Get Successor of selected node

To Preorder Array

To Inorder Array (To Sorted Array)

To Postorder Array

```
graph TD
    80((80)) --> 75((75))
    75 --> 50((50))
```

Graphic elements

There are listed all graphic elements used in this application and their meanings.

Graphic	Meaning	Description
	Start animation	Starts the animation of the tree operations.
	Pause animation	Pauses the animation of the tree operations.
	Reset animation	Resets the animation to the initial state.
	Step forward	Advances the animation by one step.
	Step back	Retracts the animation by one step.
	Continuous	Enables continuous animation.

Ketik di sini untuk mencari

11:49 21/01/2021

Balance Factor = 1

BinaryTreeVisualizer - Binary Search Tree

Not secure | btv.melezinek.cz/binary-search-tree.html

Animation

Speed of move:

Duration of a step:

Continuously ☒

History

```
insert(value: 60)
insert(value: 50)
insert(value: 75)
delete(value: 60, index: 4)
delete(value: 30, index: 1)
delete(value: 75, index: 1)
```

Algorithms

Random BSTree min:  max:

Insert value:

Find value:

Delete selected node

Get Max of selected (sub)tree

Get Min of selected (sub)tree

Get Predecessor of selected node

Get Successor of selected node

To Preorder Array

To Inorder Array (To Sorted Array)

To Postorder Array

```
graph TD
    80((80)) --> 50((50))
```

Graphic elements

There are listed all graphic elements used in this application and their meanings.

Graphic	Meaning	Description
	Start animation	Starts the animation of the tree operations.
	Pause animation	Pauses the animation of the tree operations.
	Reset animation	Resets the animation to the initial state.
	Step forward	Advances the animation by one step.
	Step back	Retracts the animation by one step.
	Continuous	Enables continuous animation.

Ketik di sini untuk mencari

11:50 21/01/2021

Balance Factor = 0

Insert 65, 30, 35

BinaryTreeVisualizer - Binary Search Tree

Not secure | btv.melezinek.cz/binary-search-tree.html

Animation

Speed of move:

Duration of a step:

Continuously ☒

History

- delete(value: 60, index: 4)
- delete(value: 30, index: 1)
- delete(value: 75, index: 1)
- insert(value: 65)
- insert(value: 30)
- delete(value: 30, index: 3)

Algorithms

Random BSTree min: 0 max: 99

Insert value:

Find value:

Delete selected node

Get Max of selected (sub)tree

Get Min of selected (sub)tree

Get Predecessor of selected node

Get Successor of selected node

To Preorder Array

To Inorder Array (To Sorted Array)

To Postorder Array

```

graph TD
    80((80)) --> 50((50))
    80 --> 65((65))
  
```

Graphic elements

There are listed all graphic elements used in this application and their meanings.

Graphic Meaning Description

Ketik di sini untuk mencari

11:51 21/01/2021

Balance Factor = 1

BinaryTreeVisualizer - Binary Search Tree

Not secure | btv.melezinek.cz/binary-search-tree.html

Animation

Speed of move:

Duration of a step:

Continuously ☒

History

- delete(value: 30, index: 1)
- delete(value: 75, index: 1)
- insert(value: 65)
- insert(value: 30)
- delete(value: 30, index: 3)
- insert(value: 30)

Algorithms

Random BSTree min: 0 max: 99

Insert value:

Find value:

Delete selected node

Get Max of selected (sub)tree

Get Min of selected (sub)tree

Get Predecessor of selected node

Get Successor of selected node

To Preorder Array

To Inorder Array (To Sorted Array)

To Postorder Array

```

graph TD
    80((80)) --> 50((50))
    80 --> 65((65))
    50 --> 30((30))
    50 --> 65
  
```

Graphic elements

There are listed all graphic elements used in this application and their meanings.

Graphic Meaning Description

Ketik di sini untuk mencari

11:51 21/01/2021

Balance Factor = 1

BinaryTreeVisualiser - Binary Search Tree

Not secure | btv.melezinek.cz/binary-search-tree.html

Animation

Speed of move:

Duration of a step:

Continuously ☒

History

delete(value: 75, index: 1)  
insert(value: 65)  
insert(value: 30)  
delete(value: 30, index: 3)  
insert(value: 30)  
insert(value: 35)

Algorithms

Random BSTree min: 0 max: 99

Insert value:

Find value:

Delete selected node

Get Max of selected (sub)tree

Get Min of selected (sub)tree

Get Predecessor of selected node

Get Successor of selected node

To Preorder Array

To Inorder Array (To Sorted Array)

To Postorder Array

```

graph TD
    80((80)) --> 50((50))
    50 --> 30((30))
    50 --> 65((65))
    30 --> 35((35))
  
```

Graphic elements

There are listed all graphic elements used in this application and their meanings.

Graphic Meaning Description

Ketik di sini untuk mencari

11:51 21/01/2021

Balance Factor = 1

Delete 80, 65, 35

BinaryTreeVisualiser - Binary Search Tree

Not secure | btv.melezinek.cz/binary-search-tree.html

Animation

Speed of move:

Duration of a step:

Continuously ☒

History

insert(value: 65)  
insert(value: 30)  
delete(value: 30, index: 3)  
insert(value: 30)  
insert(value: 35)  
delete(value: 80, index: 0)

Algorithms

Random BSTree min: 0 max: 99

Insert value:

Find value:

Delete selected node

Get Max of selected (sub)tree

Get Min of selected (sub)tree

Get Predecessor of selected node

Get Successor of selected node

To Preorder Array

To Inorder Array (To Sorted Array)

To Postorder Array

```

graph TD
    50((50)) --> 30((30))
    50 --> 65((65))
    30 --> 35((35))
  
```

Graphic elements

There are listed all graphic elements used in this application and their meanings.

Graphic Meaning Description

Ketik di sini untuk mencari

11:52 21/01/2021

Balance Factor = 1

BinaryTreeVisualizer - Binary Search Tree

Not secure | btv.melezinek.cz/binary-search-tree.html

Animation

Speed of move:

Duration of a step:

Continuously ☒

History

- insert(value: 30)
- delete(value: 30, index: 3)
- insert(value: 30)
- insert(value: 35)
- delete(value: 80, index: 0)
- delete(value: 65, index: 2)

Algorithms

Random BSTree min: 0 max: 99

Insert value:

Find value:

Delete selected node

Get Max of selected (sub)tree

Get Min of selected (sub)tree

Get Predecessor of selected node

Get Successor of selected node

To Preorder Array

To Inorder Array (To Sorted Array)

To Postorder Array

```

graph TD
    50((50)) --> 30((30))
    50 --> 35((35))
  
```

Graphic elements

There are listed all graphic elements used in this application and their meanings.

Graphic	Meaning	Description

Ketik di sini untuk mencari

11:52 21/01/2021

Balance Factor = 1

BinaryTreeVisualizer - Binary Search Tree

Not secure | btv.melezinek.cz/binary-search-tree.html

Animation

Speed of move:

Duration of a step:

Continuously ☒

History

- delete(value: 30, index: 3)
- insert(value: 30)
- insert(value: 35)
- delete(value: 80, index: 0)
- delete(value: 65, index: 2)
- delete(value: 35, index: 4)

Algorithms

Random BSTree min: 0 max: 99

Insert value:

Find value:

Delete selected node

Get Max of selected (sub)tree

Get Min of selected (sub)tree

Get Predecessor of selected node

Get Successor of selected node

To Preorder Array

To Inorder Array (To Sorted Array)

To Postorder Array

```

graph TD
    50((50)) --> 30((30))
  
```

Graphic elements

There are listed all graphic elements used in this application and their meanings.

Graphic	Meaning	Description

Ketik di sini untuk mencari

11:52 21/01/2021

Balance Factor = 0