

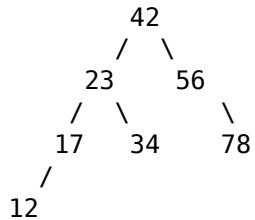
Binary Search Tree

Jacques Mock Schindler

Invalid Date

Ein binärer Suchbaum (Binary Search Tree, BST) ist eine Datenstruktur, die es ermöglicht, Daten in einer hierarchischen Struktur zu speichern. Jeder Knoten im Baum hat maximal zwei Kinder, wobei das linke Kind kleiner und das rechte Kind grösser als der Knoten selbst ist. Dies ermöglicht das effiziente Suchen, Einfügen und Löschen von Elementen.

Sollen beispielsweise die Zahlen 42, 23, 17, 34, 56, 78 und 12 der Reihe nach in einen binären Suchbaum eingefügt werden, geschieht dies wie folgt:



Um einen Knoten zu löschen, gibt es drei Fälle zu beachten: 1. Der Knoten ist ein Blatt (keine Kinder): Der Knoten kann einfach entfernt werden. 2. Der Knoten hat ein Kind: Das Kind ersetzt den Knoten. 3. Der Knoten hat zwei Kinder: Der Knoten wird durch den kleinsten Knoten im rechten Teilbaum ersetzt.

In den folgenden Abschnitten findet sich eine Mögliche Implementierung eines binären Suchbaums in Python.

Klasse BSTNode

```
1 \KeywordTok{class}\NormalTok{ BSTNode:}
2   \KeywordTok{def}\NormalTok{
3     \FunctionTok{\_\_init\_\_}\NormalTok{(\_self\NormalTok{)}
4       \NormalTok{\_self.\_key = value\NormalTok{)
5       \_self.\_parent = None\NormalTok{)
6     \_self.\_left = None\NormalTok{)
7     \_self.\_right = None\NormalTok{)
8   }
9   \NormalTok{def insert\NormalTok{(self, key, value\NormalTok{)}
10    if self.\_key == key\NormalTok{)
11      self.\_value = value\NormalTok{)
12    else if key < self.\_key\NormalTok{)
13      if self.\_left is None\NormalTok{)
14        self.\_left = BSTNode(key, value\NormalTok{)
15      else
16        self.\_left.insert(key, value\NormalTok{)
17    else if key > self.\_key\NormalTok{)
18      if self.\_right is None\NormalTok{)
19        self.\_right = BSTNode(key, value\NormalTok{)
20      else
21        self.\_right.insert(key, value\NormalTok{)
22    }
23  }
24  \NormalTok{def find\NormalTok{(self, key\NormalTok{)}
25    if self.\_key == key\NormalTok{)
26      return self.\_value\NormalTok{)
27    else if key < self.\_key\NormalTok{)
28      if self.\_left is None\NormalTok{)
29        return None\NormalTok{)
30      else
31        return self.\_left.find(key\NormalTok{)
32    else if key > self.\_key\NormalTok{)
33      if self.\_right is None\NormalTok{)
34        return None\NormalTok{)
35      else
36        return self.\_right.find(key\NormalTok{)
37    }
38  }
39  \NormalTok{def delete\NormalTok{(self, key\NormalTok{)}
40    if self.\_key == key\NormalTok{)
41      if self.\_left is None\NormalTok{)
42        if self.\_right is None\NormalTok{)
43          self.\_parent.\_left = None\NormalTok{)
44        else
45          self.\_right.\_parent = self.\_parent\NormalTok{)
46          self.\_parent.\_left = self.\_right\NormalTok{)
47      else if self.\_right is None\NormalTok{)
48        self.\_parent.\_left = self.\_left\NormalTok{)
49        self.\_left.\_parent = self.\_parent\NormalTok{)
50      else
51        min_right = self.\_right
52        while min_right.\_left is not None\NormalTok{)
53          min_right = min_right.\_left\NormalTok{)
54        min_right.\_parent = self.\_parent\NormalTok{)
55        self.\_parent.\_left = min_right\NormalTok{)
56        min_right.\_left = self.\_left\NormalTok{)
57        min_right.\_right = self.\_right\NormalTok{)
58    else if key < self.\_key\NormalTok{)
59      if self.\_left is not None\NormalTok{)
60        self.\_left.delete(key\NormalTok{)
61      else
62        print("Key not found in left subtree")\NormalTok{)
63    else if key > self.\_key\NormalTok{)
64      if self.\_right is not None\NormalTok{)
65        self.\_right.delete(key\NormalTok{)
66      else
67        print("Key not found in right subtree")\NormalTok{)
68    }
69  }
70  \NormalTok{def in_order\NormalTok{(self\NormalTok{)}
71    if self.\_left is not None\NormalTok{)
72      self.\_left.in_order\NormalTok{)
73    print(self.\_key, self.\_value\NormalTok{)
74    if self.\_right is not None\NormalTok{)
75      self.\_right.in_order\NormalTok{)
76  }
77  \NormalTok{def pre_order\NormalTok{(self\NormalTok{)}
78    print(self.\_key, self.\_value\NormalTok{)
79    if self.\_left is not None\NormalTok{)
80      self.\_left.pre_order\NormalTok{)
81    if self.\_right is not None\NormalTok{)
82      self.\_right.pre_order\NormalTok{)
83  }
84  \NormalTok{def post_order\NormalTok{(self\NormalTok{)}
85    if self.\_left is not None\NormalTok{)
86      self.\_left.post_order\NormalTok{)
87    if self.\_right is not None\NormalTok{)
88      self.\_right.post_order\NormalTok{)
89    print(self.\_key, self.\_value\NormalTok{)
90  }
91}
```

```

6      \VariableTok{self}\NormalTok{.left }\operatorTok{=} \\
7 \VariableTok{None} \\
8     \VariableTok{self}\NormalTok{.right }\operatorTok{=} \\
9 \VariableTok{None} \\
10    \KeywordTok{def} \\
11 \FunctionTok{\_\_str\_\_}\NormalTok{()}\VariableTok{self}\NormalTok{.key}) \\
12 \NormalTok{key }\operatorTok{=} \\
13 \BuiltInTok{str}\NormalTok{()}\VariableTok{self}\NormalTok{.key}) \\
14 \NormalTok{parent }\operatorTok{=} \\
15 \StringTok{\text{if}}\NormalTok{ }\VariableTok{self}\NormalTok{.parent }\KeywordTok{is} \VariableTok{None} \\
16 \ControlFlowTok{else} \\
17 \BuiltInTok{str}\NormalTok{()}\VariableTok{self}\NormalTok{.parent.key}) \\
18 \NormalTok{left }\operatorTok{=} \\
19 \StringTok{\text{if}}\NormalTok{ }\VariableTok{self}\NormalTok{.left }\KeywordTok{is} \VariableTok{None} \\
\ControlFlowTok{else} \\
\BuiltInTok{str}\NormalTok{()}\VariableTok{self}\NormalTok{.left.key}) \\
\NormalTok{right }\operatorTok{=} \\
\StringTok{\text{if}}\NormalTok{ }\VariableTok{self}\NormalTok{.right }\KeywordTok{is} \VariableTok{None} \\
\ControlFlowTok{else} \\
\BuiltInTok{str}\NormalTok{()}\VariableTok{self}\NormalTok{.right.key}) \\
\NormalTok{s }\operatorTok{=} \\
\SpecialStringTok{f\text{'}}\NormalTok{ }\CharTok{\textbackslash} \\
t}\SpecialStringTok{Parent =} \\
}\SpecialCharTok{\{}\NormalTok{parent}\SpecialCharTok{\}}\CharTok{\textbackslash} \\
slash\NormalTok{ }\SpecialStringTok{f\text{'}}\NormalTok{ }\CharTok{\textbackslash} \\
t}\SpecialStringTok{Key =} \\
}\SpecialCharTok{\{}\NormalTok{key}\SpecialCharTok{\}}\CharTok{\textbackslash} \\
slash\NormalTok{ }\SpecialStringTok{f\text{'}}\NormalTok{ }\CharTok{\textbackslash} \\
t}\SpecialStringTok{Left =} \\
}\SpecialCharTok{\{}\NormalTok{left}\SpecialCharTok{\}}\CharTok{\textbackslash} \\
slash\NormalTok{ }\SpecialStringTok{Right =} \\
}\SpecialCharTok{\{}\NormalTok{right}\SpecialCharTok{\}}\SpecialStringTok{\{}\text{'}}\NormalTok{ }\CharTok{\textbackslash} \\
text\NormalTok{ }\SpecialStringTok{f\text{'}}\NormalTok{ }\CharTok{\textbackslash} \\
t}
\NormalTok{) } \\
\ControlFlowTok{return}\NormalTok{ s }

```

Klasse BST

```

1 \KeywordTok{class}\NormalTok{ BST:} \\
2   \KeywordTok{def} \\
3 \FunctionTok{\_\_init\_\_}\NormalTok{()}\VariableTok{self}\NormalTok{.key}\operatorTok{=} \\
4 \NormalTok{None}\NormalTok{.value}\operatorTok{=} \\
5 \NormalTok{None} \\
6 \NormalTok{None}\operatorTok{=} \\
7 \NormalTok{None}

```

```

3      \ControlFlowTok{if}\NormalTok{ key }\KeywordTok{is}
4      \VariableTok{None}\NormalTok{:\}}
5          \VariableTok{self}\NormalTok{.root }\OperatorTok{=}
6      \VariableTok{None}
7          \ControlFlowTok{else}\NormalTok{:\}
8      \NormalTok{ node }\OperatorTok{=}\NormalTok{ BSTNode(key, value)
9          \VariableTok{self}\NormalTok{.root }\OperatorTok{=}\NormalTok{ node}\NormalTok{:\}
10     \NormalTok{ node }\OperatorTok{=}\NormalTok{ BSTNode(key, value)}
11     \ControlFlowTok{if} \VariableTok{self}\NormalTok{.root
12 } \KeywordTok{is} \VariableTok{None}\NormalTok{:\}
13     \VariableTok{self}\NormalTok{.root }\OperatorTok{=}\NormalTok{ node}\NormalTok{:\}
14     \ControlFlowTok{return}
15
16     \ControlFlowTok{if}\NormalTok{ root }\KeywordTok{is}
17     \VariableTok{None}\NormalTok{:\}
18     \NormalTok{ root }\OperatorTok{=}
19     \VariableTok{self}\NormalTok{.root}
20
21     \ControlFlowTok{if}\NormalTok{ key
22     }\OperatorTok{\textless}\{}\NormalTok{ root.key }\KeywordTok{and}\NormalTok{ root.left }\KeywordTok{is}
23     \NormalTok{ root.left }\OperatorTok{=}\NormalTok{ node}\NormalTok{:\}
24     \NormalTok{ root.left }\OperatorTok{=}\NormalTok{ node.parent }\OperatorTok{=}\NormalTok{ root}\NormalTok{:\}
25     \ControlFlowTok{return}
26
27
28     \ControlFlowTok{if}\NormalTok{ key
29     }\OperatorTok{\textless}\{}\NormalTok{ root.key:\}
30     \NormalTok{ root }\OperatorTok{=}\NormalTok{ root.left}\NormalTok{:\}
31     \VariableTok{self}\NormalTok{.insert(key, value, root)}
32
33
34     \ControlFlowTok{if}\NormalTok{ key
35     }\OperatorTok{\textgreater}\{}\NormalTok{ root.key:\}
36     \NormalTok{ root }\OperatorTok{=}\NormalTok{ root.right}\NormalTok{:\}
37     \VariableTok{self}\NormalTok{.insert(key, value, root)}

```

```

37     \KeywordTok{def}
38     \BuiltInTok{min}\NormalTok{()}\VariableTok{self}\NormalTok{, bst}\OperatorTok{=}\VariableTok{None}\NormalTok{():}
39         \ControlFlowTok{if}\NormalTok{ bst }\KeywordTok{is}
40         \VariableTok{None}\NormalTok{::}
41         \NormalTok{minimum }\OperatorTok{=}
42         \VariableTok{self}\NormalTok{.root}
43         \ControlFlowTok{else}\NormalTok{::}
44         \NormalTok{minimum }\OperatorTok{=}\NormalTok{bst.root}
45
46         \ControlFlowTok{while}\NormalTok{ minimum.left }\KeywordTok{is}
47         \KeywordTok{not} \VariableTok{None}\NormalTok{::}
48         \NormalTok{minimum }\OperatorTok{=}\NormalTok{ minimum.left}
49
50     \KeywordTok{def}
51     \BuiltInTok{max}\NormalTok{()}\VariableTok{self}\NormalTok{, bst}\OperatorTok{=}\VariableTok{None}\NormalTok{():}
52         \ControlFlowTok{if}\NormalTok{ bst }\KeywordTok{is}
53         \VariableTok{None}\NormalTok{::}
54         \NormalTok{maximum }\OperatorTok{=}
55         \VariableTok{self}\NormalTok{.root}
56         \ControlFlowTok{else}\NormalTok{::}
57         \NormalTok{maximum }\OperatorTok{=}\NormalTok{bst.root}
58
59         \ControlFlowTok{while}\NormalTok{ maximum.right }\KeywordTok{is}
60         \KeywordTok{not} \VariableTok{None}\NormalTok{::}
61         \NormalTok{maximum }\OperatorTok{=}\NormalTok{ maximum.right}
62
63         \ControlFlowTok{return}\NormalTok{ maximum}
64
65     \KeywordTok{def}\NormalTok{ search()\VariableTok{self}\NormalTok{, key, node}\OperatorTok{=}\VariableTok{None}\NormalTok{():}
66         \CommentTok{\# If initial call or we've hit None in recursion}
67         \ControlFlowTok{if}\NormalTok{ node }\KeywordTok{is}
68         \VariableTok{None}\NormalTok{::}
69             \ControlFlowTok{if}\NormalTok{ \VariableTok{self}\NormalTok{.root }
70                 \KeywordTok{is} \VariableTok{None}\NormalTok{::}
71                     \CommentTok{\# Empty tree}
72                     \ControlFlowTok{return}\NormalTok{ DecValTok{1}}
73             \NormalTok{node }\OperatorTok{=}\NormalTok{ \VariableTok{self}\NormalTok{.root}
74
75             \CommentTok{\# Found the key}
76             \ControlFlowTok{if}\NormalTok{ key }\OperatorTok{==}\NormalTok{ node.key}
77                 \ControlFlowTok{return}\NormalTok{ node}
78
79             \CommentTok{\# Key doesn't exist in this path}
80             \ControlFlowTok{if}\NormalTok{ key }\OperatorTok{less}\NormalTok{ node.key}
81                 \NormalTok{node.key:=}

```

```

72          \ControlFlowTok{if}\NormalTok{ node.left }\KeywordTok{is}
73      \VariableTok{None}\NormalTok{:\}}
74          \ControlFlowTok{return} \OperatorTok{{-}}\DecValTok{1}
75          \ControlFlowTok{return}
76      \VariableTok{self}\NormalTok{.\search(key, node.left)}
77          \ControlFlowTok{else}\NormalTok{:\# key}
78          \textgreater{}\node.key}
79          \ControlFlowTok{if}\NormalTok{ node.right }\KeywordTok{is}
80      \VariableTok{None}\NormalTok{:\}
81          \ControlFlowTok{return} \OperatorTok{{-}}\DecValTok{1}
82          \ControlFlowTok{return}
83      \VariableTok{self}\NormalTok{.\search(key, node.right)}
84
85          \KeywordTok{def}\NormalTok{ delete()\VariableTok{self}\NormalTok{.\NormalTok{key:}}
86          \CommentTok{\# Find the node to delete}
87      \NormalTok{node }\OperatorTok{=}
88      \VariableTok{self}\NormalTok{.\search(key)}
89
90          \CommentTok{\# If node not found, return}
91          \ControlFlowTok{if}\NormalTok{ node }\OperatorTok{==}
92          \OperatorTok{{-}}\DecValTok{1}\NormalTok{:\}
93          \ControlFlowTok{return}
94
95          \VariableTok{self}\NormalTok{.\_delete\_node(node)}
96
97          \KeywordTok{def}\NormalTok{_delete()\VariableTok{self}\NormalTok{.\NormalTok{node:}}
98          \CommentTok{\# Case 1: Node has no children (leaf node)}
99          \ControlFlowTok{if}\NormalTok{ node.left }\KeywordTok{is}
100         \VariableTok{None}\NormalTok{ \KeywordTok{and}\NormalTok{ node.right }\KeywordTok{is}
101         \VariableTok{None}\NormalTok{:\}
102         \ControlFlowTok{if}\NormalTok{ node.\operatorname{parent}.left }
103         \OperatorTok{==}\NormalTok{ node:\}
104         \NormalTok{node.parent.left }\OperatorTok{=}
105         \VariableTok{None}\NormalTok{:\}
106         \ControlFlowTok{else}\NormalTok{:\}
107         \ControlFlowTok{if}\NormalTok{ node.parent.left }
108         \OperatorTok{==}\NormalTok{ node:\}
109         \NormalTok{node.parent.left }\OperatorTok{=}
110         \VariableTok{None}\NormalTok{:\}
111         \ControlFlowTok{else}\NormalTok{:\}
112         \NormalTok{node.parent.right }\OperatorTok{=}
113         \VariableTok{None}\NormalTok{:\}
114
115         \CommentTok{\# Case 2: Node has only one child}
116         \ControlFlowTok{elif}\NormalTok{ node.left }\KeywordTok{is}
117         \VariableTok{None}\NormalTok{:\# Has only right child}
118         \ControlFlowTok{if}\NormalTok{ node.\operatorname{parent}.right }
119         \OperatorTok{==}\NormalTok{ node:\}
120         \VariableTok{self}\NormalTok{.\operatorname{root}:}
121         \VariableTok{self}\NormalTok{.\operatorname{root}
122         \OperatorTok{=}\NormalTok{ node.right}

```

```

105 \NormalTok{          node.right.parent }\operatorTok{=}
106 \VariableTok{None}      \ControlFlowTok{else}\NormalTok{:}
107             \ControlFlowTok{if}\NormalTok{ node.parent.left
108 }\operatorTok{==}\NormalTok{ node:\node.parent.left }\operatorTok{=}\NormalTok{ node.parent.left
109 \NormalTok{          node.parent.left }\operatorTok{=}\NormalTok{ node.parent.right
110 \NormalTok{          node.parent.right }\operatorTok{=}\NormalTok{ node.parent.right
111 \NormalTok{          node.parent }\operatorTok{=}\NormalTok{ node.parent
112
113             \ControlFlowTok{else}\NormalTok{:}
114 \VariableTok{None}\NormalTok{: }\CommentTok{\# Has only left child}
115             \ControlFlowTok{if}\NormalTok{ node }\operatorTok{==}
116 \VariableTok{self}\NormalTok{.root:\node.parent.root }
117             \VariableTok{self}\NormalTok{.root
118 }\operatorTok{=}\NormalTok{ node.left }\operatorTok{=}\NormalTok{ node.left.parent
119 \NormalTok{          node.left.parent }\operatorTok{=}
120 \VariableTok{None}      \ControlFlowTok{else}\NormalTok{:}
121             \ControlFlowTok{if}\NormalTok{ node.parent.left
122 }\operatorTok{==}\NormalTok{ node:\node.parent.left }\operatorTok{=}\NormalTok{ node.parent.left
123 \NormalTok{          node.parent.left }\operatorTok{=}\NormalTok{ node.parent.right
124 \NormalTok{          node.parent.right }\operatorTok{=}\NormalTok{ node.parent.right
125 \NormalTok{          node.parent.right }\operatorTok{=}\NormalTok{ node.parent.parent
126
127             \CommentTok{\# Case 3: Node has two children}
128             \ControlFlowTok{else}\NormalTok{:}
129                 \CommentTok{\# Find successor (smallest node in right subtree)}
130 \NormalTok{          successor }\operatorTok{=}\NormalTok{ \VariableTok{None}
131 \NormalTok{          current }\operatorTok{=}\NormalTok{ node.right
132
133             \ControlFlowTok{while}\NormalTok{ current.left }\operatorTok{=}\NormalTok{ current
134                 \CommentTok{\# Copy successor\text{ quotesingle{}s key and value to
135 the node}
136 \NormalTok{          node.key }\operatorTok{=}\NormalTok{ successor.key
137 \NormalTok{          node.value }\operatorTok{=}\NormalTok{ successor.value
138
139             \CommentTok{\# Delete the successor (which has at most one right
child)}}
```

```

140         \VariableTok{self}\NormalTok{\_}\_delete\_\_node(successor)\}
141
142     \KeywordTok{def}\NormalTok{\_iterate()}\VariableTok{self}\NormalTok{\_}\_delete\_\_node(successor)\}
143     \operatorname{def} \_iterate():
144         \operatorname{if} self._node is None:
145             \operatorname{return} []
146
147         \operatorname{if} self._root is None:
148             \operatorname{return} []
149
150         \operatorname{if} self._root is None:
151             \operatorname{return} []
152
153         \operatorname{if} self._root is None:
154             \operatorname{return} []
155
156         \operatorname{if} self._root is None:
157             \operatorname{return} []
158
159         \operatorname{if} self._root is None:
160             \operatorname{return} []
161
162         \operatorname{if} self._root is None:
163             \operatorname{return} []

```