B+ Tree node properties (m = order of tree)

Root Node (Level 0)

- **Minimum number of keys**: 1 (if it's not the only node in the tree)
- **Maximum number of keys**: **m - 1**
- **Minimum number of children**: 2 (if it's not the only node in the tree)
- **Maximum number of children**: **m**

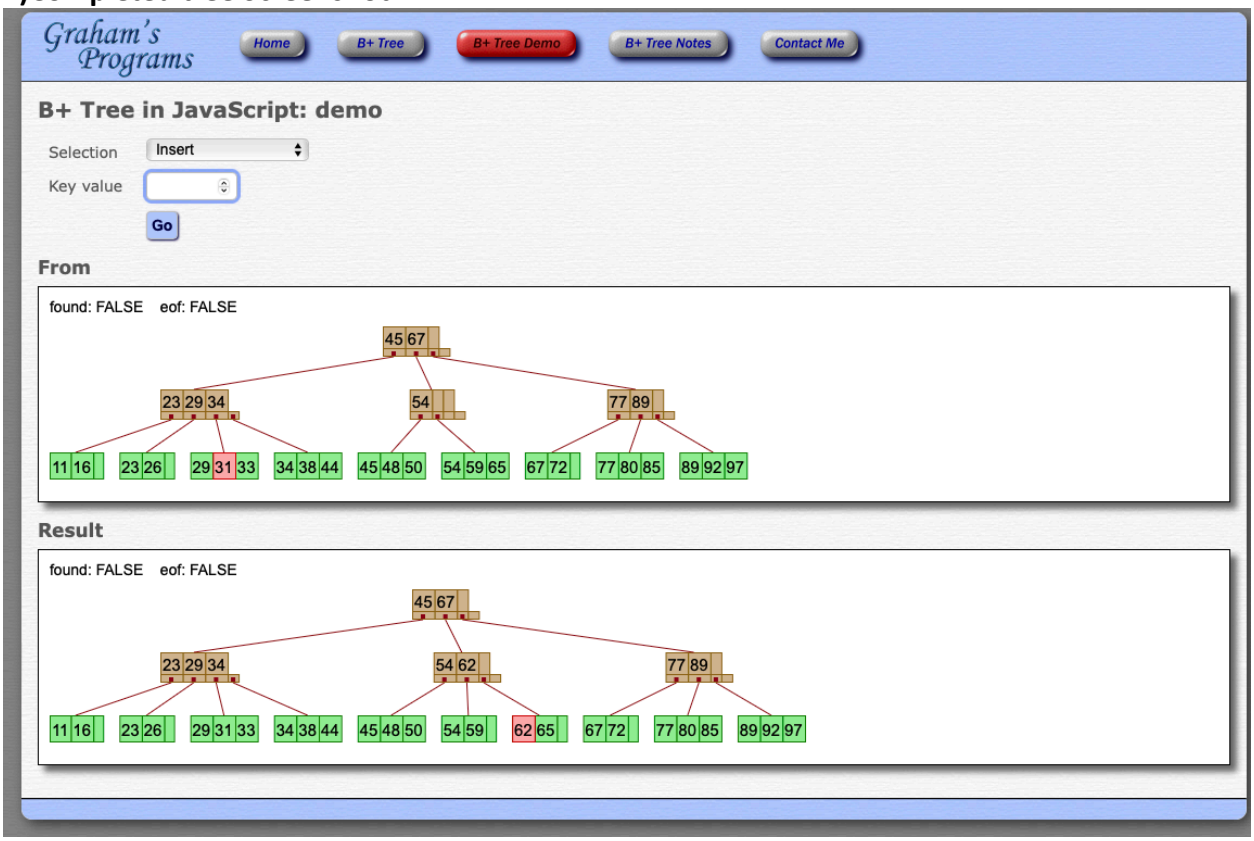
Internal Nodes (Level 1 and above, except root)

- **Minimum number of keys**: **[m/2] - 1**
- **Maximum number of keys**: **m - 1**
- **Minimum number of children**: **[m/2]**
- **Maximum number of children**: **m**

Leaf Nodes (Bottom Level)

- **Minimum number of keys**: **[m/2] - 1**
- **Maximum number of keys**: **m - 1**
- **Children**: N/A (leaf nodes do not have children, but they have pointers to the next leaf node for a linked list if it is a B+ tree)

1)Completed tree Screenshot:

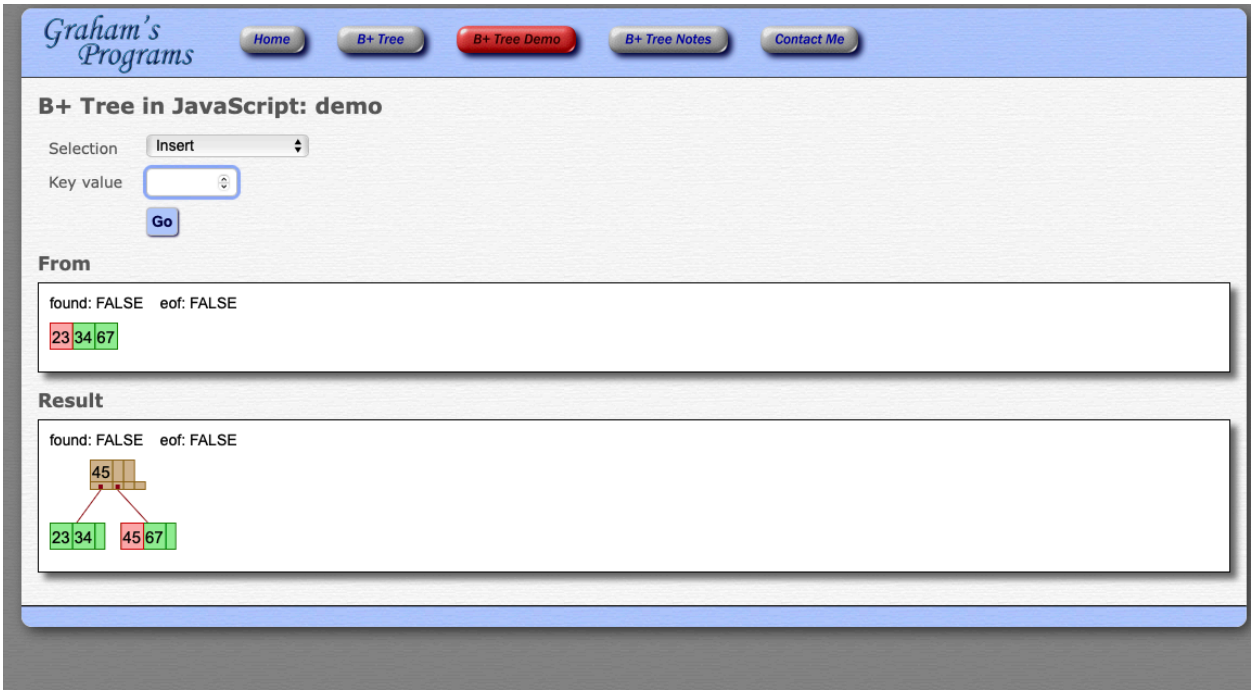


25 random values:

2)Inserting Screenshot:

Before

After

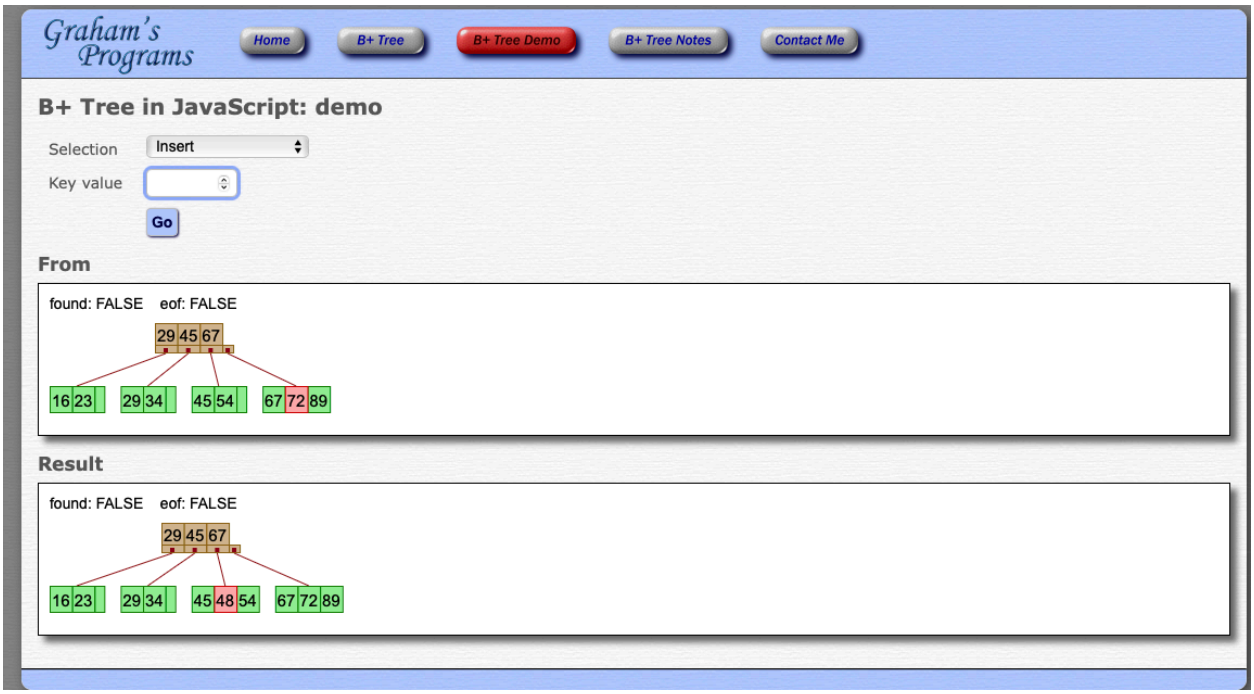


when we insert 45, the node looks like this before splitting: [23, 34, 45, 67]

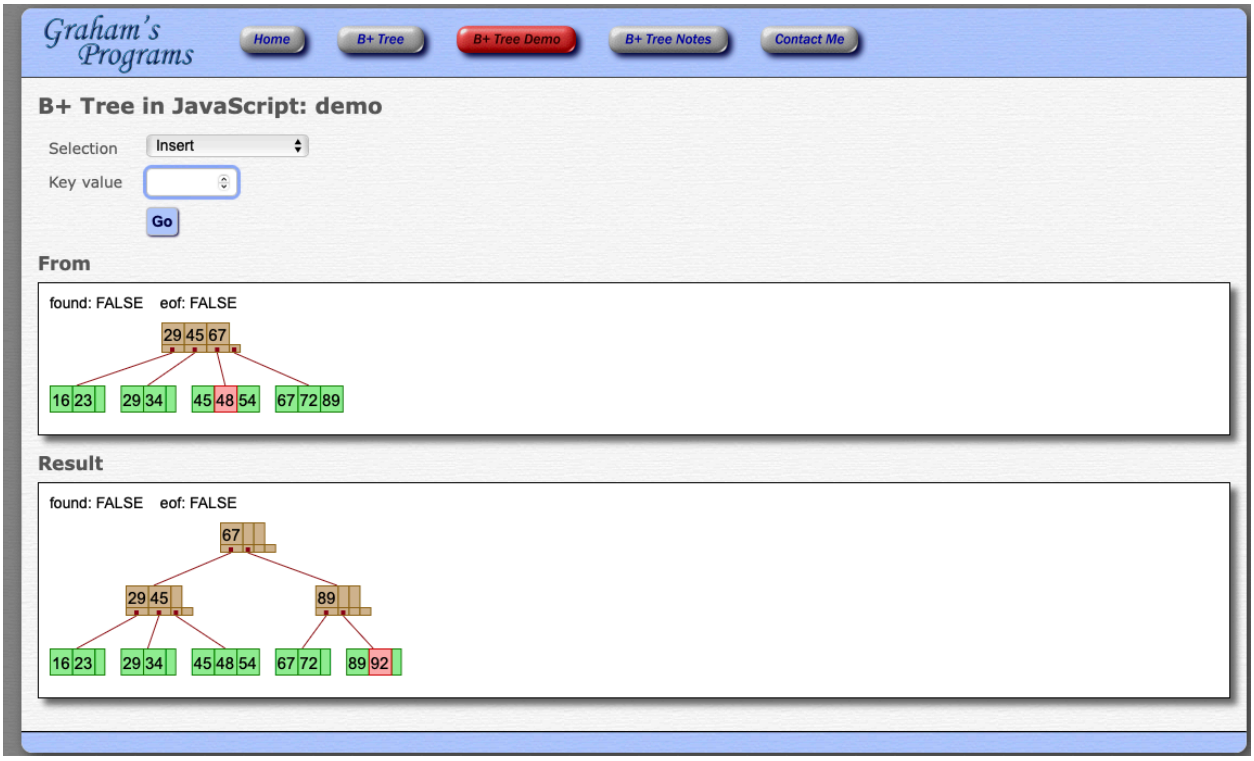
Since the node can only hold a maximum of 3 keys, the insertion of the fourth key (45) triggers a split operation. The node will split into two nodes [23,34] and [45,67]

3)Height increase Screenshot:

Before



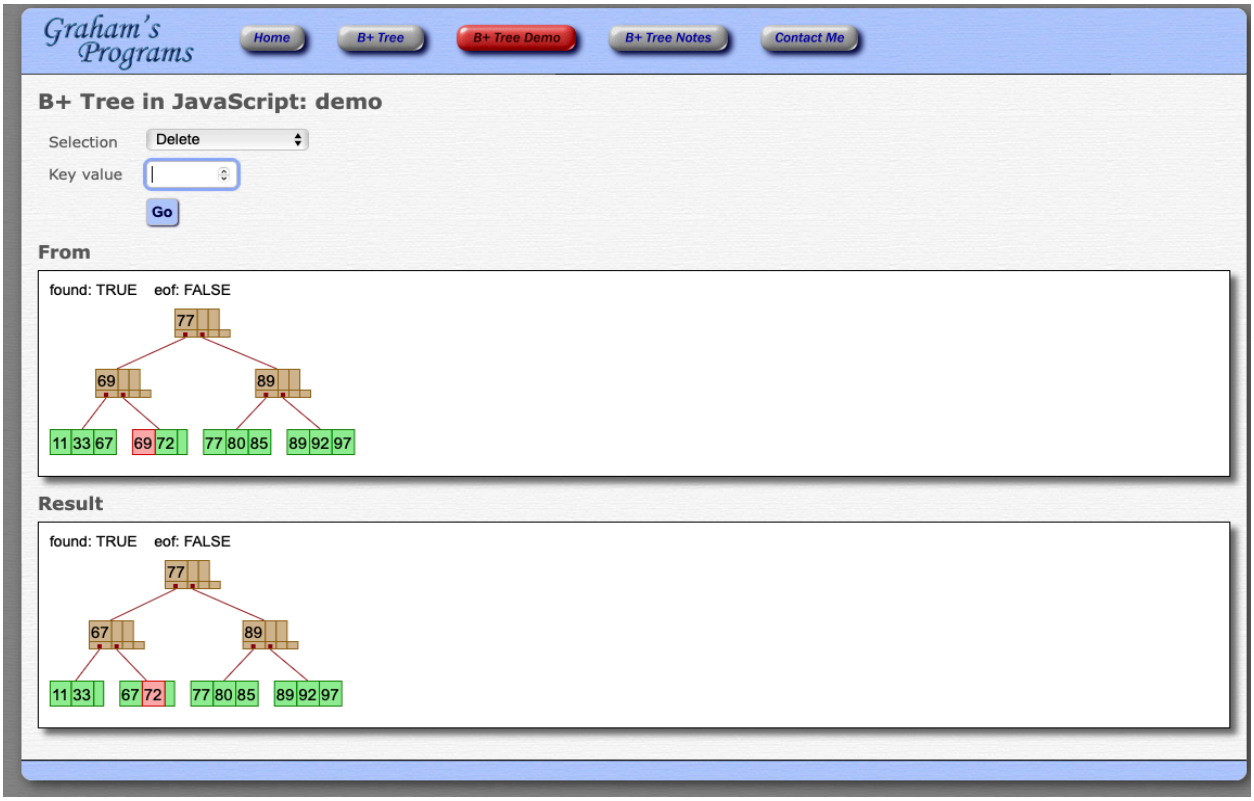
After



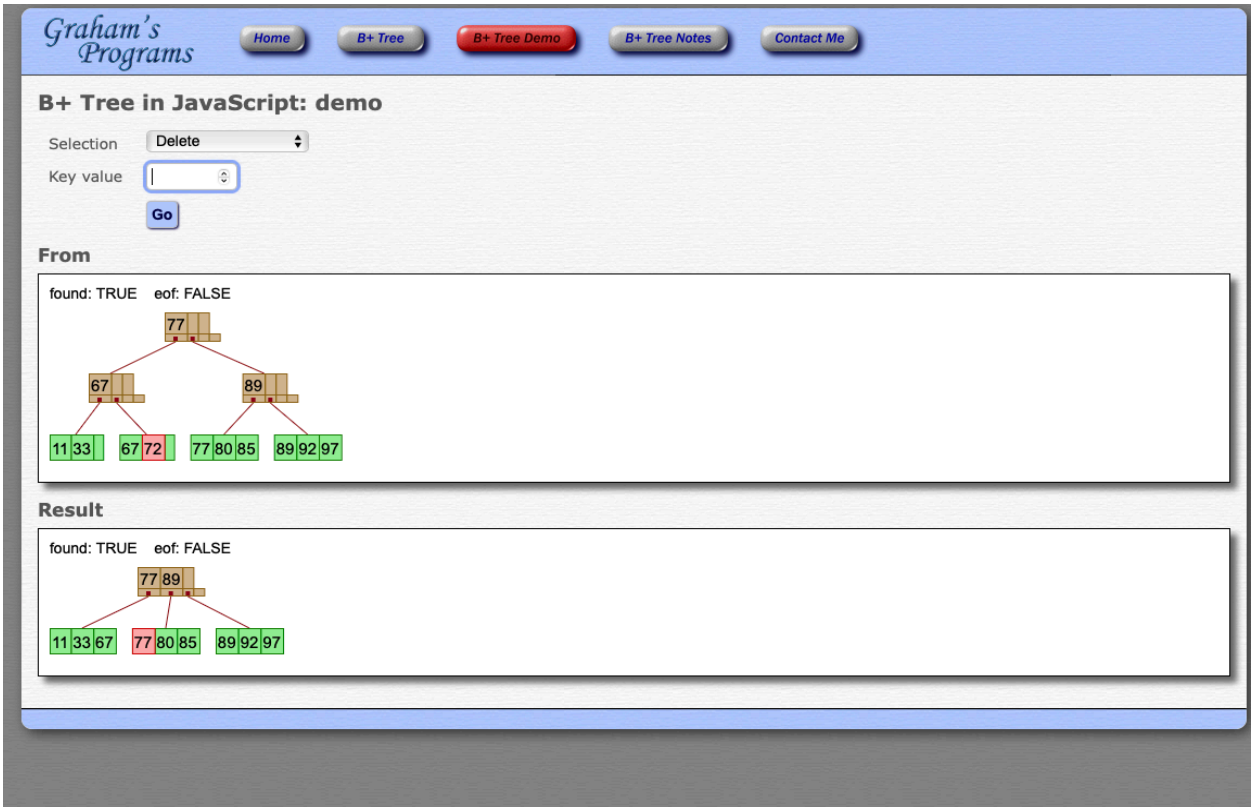
When the value 92 is inserted, it is placed into the leaf node [67 72 89], causing it to overflow. Since a B+ tree of order 4 cannot have more than 3 values in a leaf node, the node splits. The median value 72 in this case, since 67 was promoted earlier it is pushed up into the parent node. However, since there's no space in the parent node, it causes the root to split as well, and the median value of the root node (67) becomes the new root. This split is what increases the height of the tree from 2 levels to 3 levels.

4)Deletion Height Decrease Screenshot:

Before



After



The deletion of key 69 from the leaf node that initially contained [69, 72] leads to an underflow, which is against the B+ tree's property of having a minimum number of keys in non-root nodes. To correct this, the tree undergoes a restructuring. The lone key [72] merges with its adjacent sibling node [77, 80, 85], causing a redistribution of keys to maintain the B+ tree invariants. This redistribution pushes the key 77 upward into the parent node, which causes the parent node to now have [77, 89], This decreases the height of the tree from 3 levels to 2 levels.