Examples of delete operations of the binary search tree

|  |
| --- |
| **The delete operation** |
| void Delete(Node \*newTree, int value)  {  if (newTree->getKey() == -1)  {  return;  }  else if (value < newTree->getKey())  {  Delete(newTree->getLeft(), value);  }  else if (value > newTree->getKey())  {  Delete(newTree->getRight(), value);  }  else  {  if (newTree->getLeft() == NULL && newTree->getRight() == NULL)  {  newTree->delete\_node();  }  else if (newTree->getLeft() == NULL)  {  newTree->setKey(newTree->getRight()->getKey());  newTree->setRight(newTree->getRight());  }  else if (newTree->getRight() == NULL)  {  newTree->setKey(newTree->getLeft()->getKey());  newTree->setLeft(newTree->getLeft());  }  else  {  Node\* temp = findMin(newTree->getRight());  newTree->changeKey(temp->getKey());  Delete(newTree->getRight(), temp->getKey());  }  }  }  void changeKey(int key) { data = key; }  void delete\_node() { data = -1, left = NULL, right = NULL; }  Node\* findMin(Node\* newTree)  {  while (newTree->left != NULL)  {  newTree=newTree->left;  }  return newTree;  } |