Convert Sorted Linked List to Binary Search Tree [LeetCode](https://leetcode.com/problems/convert-sorted-list-to-binary-search-tree/description/)

Given the head of a singly linked list where elements are sorted in **ascending order**, convert *it to a*

***height-balanced*** *binary search tree*.

Example:

The Sorted Linked List: -10 → -3 → 0 → 5 → 9

Output:

0

/ \

-10 5

\ \

-3 9

**Approach 1: Function to start the process of converting a sorted linked list to a balanced binary search tree**

* **Function Purpose**:
  + The **sortedListToBST** function converts a sorted linked list into a balanced binary search tree.
* **Explanation**:
  + It uses a recursive approach to divide the linked list and build the tree.
  + The **solve** function is called recursively to find the middle of the linked list and create a new node for the middle value.
  + The process is repeated for the left and right halves of the linked list, creating the left and right subtrees.
  + The function returns the root of the binary search tree.
* **Time Complexity**:
  + **O(N), where N is the number of elements in the linked list, as each element is processed once.**
* **Space Complexity**:
  + **Overall space complexity is O(N) due to the memory used by the binary search tree.**