Convert Sorted Array to Balanced Binary Search Tree

```
#include <stdio.h>
#include <stdlib.h>

struct BTNode
{
	int data;
	struct BTNode *left;
	struct BTNode *right;
};

struct BTNode *newNode(int data)
{
	struct BTNode *temp = (struct BTNode *)malloc(sizeof(struct BTNode));
	temp->data = data;
	temp->left = NULL;
	temp->right = NULL;
	return temp;
}

struct BTNode *sortedArrayToBalencedBST(int *arr, int start, int end)
{
	if(start > end)
		return NULL;

	int middle = (start + end) / 2;
	struct BTNode *root = newNode(arr[middle]);

	root->left = sortedArrayToBalencedBST(arr, start, middle-1);

	root->right = sortedArrayToBalencedBST(arr, middle+1, end);

	return root;
}

void printPreOrder(struct BTNode *root)
{
	if(root)
	{
		printf("%d\t", root->data);
		printPreOrder(root->left);
		printPreOrder(root->right);
	}
}

int main()
{
	int *arr, size;
	printf("Enter size of an array\n");
	scanf("%d", &size);
```

```
    //allocate memory
    arr = (int *)malloc(size * sizeof(int));

    printf("Enter sorted array elements\n");
    for(int index = 0; index < size; index++)
        scanf("%d", &arr[index]);

    struct BTNode *root = sortedArrayToBalencedBST(arr, 0, size-1);

    //inorder traversal
    printPreOrder(root);
    return 0;
}
```

Time complexity: O(n)

Space complexity: O(n)