RAPORT OF PROJECT 2 WİTHİN CSE 2025 – DATA STRUCTURE

Harun Büyüktepe
150115020

This project was a programming assignment in C which aims to build two algorithms (BST and AVL) that will store/search/update/delete information related to textual materials and make performance experiments in order to evaluate our system.

In the project, the comparison was made between the AVL tree and the BS tree in general.Execution time and memory usage measurements were made in the project.The time kept was often different but the usage of memory was always the same.

Some Measurament of Execution Time (Only Insertetion)(2000*s)				
*	**BS Tree for Text1**	**AVL Tree for Text1**	**BS Tree for Text2**	**AVL Tree for Text2**
1.	0.94	0.124	0.94	0.110
2.	0.109	0.109	0.96	0.125
3.	0.110	0.125	0.94	0.109
4.	0.110	0.110	0.93	0.110
5.	0.109	0.125	0.94	0.109

(figure I)

We got very contradictory results .In theory the AVL tree should be faster but the AVL tree is slower because the struct of AVL tree is larger than BS tree. Actually texts are not long enough to make clear decisions. For this reason, the use of memory is more at AVL tree than BS tree in this project.

Some Measurament of Memory Usage (Only Insertetion)(1*Byte)				
*	**BS Tree for Text1**	**AVL Tree for Text1**	**BS Tree for Text2**	**AVL Tree for Text2**
1.	6061	6765	5721	6385
2.	6061	6765	5721	6385
3.	6061	6765	5721	6385
4.	6061	6765	5721	6385
5.	6061	6765	5721	6385

(figure II)

To delete words from avl tree was shorter than bs tree. In this section, the theoretical data and the experimental data are consistent.

Some Measurament of Execution Time (Insertetion with Deletion)(2000*s)								
*	**BS Tree for Text1**		**AVL Tree for Text1**		**BS Tree for Text2**		**AVL Tree for Text2**	
	İnsert only	İns./delete	İnsert only	İns./delete	İnsert only	İns./delete	İnsert only	İns./delete
1.	0.94	0.172	0.124	0.203	0.110	0.186	0.110	0.187
2.	0.109	0.187	0.140	0.204	0.109	0.172	0.125	0.188
3.	0.110	0.187	0.125	0.219	0.109	0.188	0.120	0.188
4.	0.110	0.171	0.110	0.165	0.93	0.165	0.110	0.156
5.	0.110	0.171	0.109	0.203	0.110	0.172	0.109	0.187

(figure III)

The difference magnitude of structs which are BST and AVL 's nodes and the difference between the sentences which will be deleted different number of word affect to give different memory usage values for AVL and BST.

Some Measurament of Memory Usage (Insertetion with Deletion)(1*Byte)								
*	**BS Tree for Text1**		**AVL Tree for Text1**		**BS Tree for Text2**		**AVL Tree for Text2**	
	İnsert only	İns./delete	İnsert only	İns./delete	İnsert only	İns./delete	İnsert only	İns./delete
1.	6061	4997	6765	5549	5721	4713	6385	5233
2.	6061	4997	6765	5549	5721	4713	6385	5233
3.	6061	4997	6765	5549	5721	4713	6385	5233
4.	6061	4997	6765	5549	5721	4713	6385	5233
5.	6061	4997	6765	5549	5721	4713	6385	5233

(figure IV)

In shortly,we obtain that AVL Tree brings useful , fast search processing.BS tree also can use at short data processing to give maximum efficiency but if we will work on big projects , we have to use AVL Tree to give maximum efficieny.

Harun Büyüktepe
150115020