Programming Assignment #1-REPORT

B05901155　李羚毓

Structures

- node
  Contains the key, the left child, the right child, and the order when visited by inorder traversal of a node.
- node_linked_list
  Contains the key and the next node linked list of a node_linked_list, and an integer recording whether the next node_linked exists.

Functions

*n: the number of elements

- S_node* Create_node();
  Creates a node.
  Time complexity: O(1)
- void Copy_node(struct node*, struct node*);
  Copies the elements of the second parameter into the first parameter.
  Time complexity: O(1)
- void Standard_push(struct node*, int);
  Inserts an element into the standard tree.
  Average time complexity: O(log n)
- int Splay_push(struct node*, int);
  Inserts an element into the splay tree.
  Average time complexity: O(log n)
- void Left_rotate(struct node*);
  Left rotation of splay tree.
  Time complexity: O(1)
- void Right_rotate(struct node*);
  Right rotation of splay tree.
  Time complexity: O(1)
- int Print_line(struct node*, struct node*, int, FILE*);
  Prints the parenthesis presentations of a tree into the third parameter.
  Realized by DFS.
  Time complexity: O(n)
- void Print_tree_and_left_boundary(struct node*, FILE*, FILE*);
  Prints the textual printing of a tree into the second parameter, and the left boundaries of a tree into the third parameter.
  First calls the function “Set_priority_and_left_boundary”, then the textual

printing of a tree is done by BFS with the orders and the left boundaries known.
Time complexity: O(n)

- void Set_priority_and_left_boundary(struct node*, int*, S_node_linked_list*);
Finds the order visited by inorder traversal and the left boundaries of a tree.
Realized by DFS.
Time complexity: O(n)
- S_node_linked_list* Create_node_linked_list(struct node*);
Creates a node_linked_list.
Time complexity: O(1)
- void Spacing(int, FILE*);
Prints spaces into a file.
Time complexity: O(the largest number of spaces to be printed)
- int Num_length(int);
Calculates the number of digits of a number.
Time complexity: O(the largest possible number of digits of a number)
- void Destory_tree(struct node*);
Free the spaces of the tree.
Realized by DFS.
Time complexity: O(n)

Space Complexity Analysis

*n: the number of elements

Space Complexity: O(n)

The spaces of nodes and node_linked_lists are allocated only when there is a key to be stored, and are freed after there are no later use.

Bibliography

Lecture slides.