BLG335E - Analysis of Algorithms I
2020-2021 Fall
Homework-4

10.02.2021
Res. Asst. Yusuf H. Sahin
sahinyu@itu.edu.tr

- You should write all your code in C++ language. You should check your code using the shared HackerRank pages. Your outputs should in the same format with the outputs given in the HackerRank. **Your code will be evaluated using HackerRank system. Thus, no partial points will be given for codes with no matching outputs.**
- Please write your own codes, copying code parts from books, websites or any other source including your friends is considered as plagiarism and results in penalty. Also disciplinary actions will be taken.
- Do not upload your codes to any public platform (e.g. Github) until the deadline of homework passes.
- Submit your source codes and report files on Ninova before the deadline, late submissions and submissions via e-mail will not be accepted.
- If you have any questions, please use the message board.
- **You should both upload your project to HackerRank and Ninova.**
- Use your real name/ITU mail in HackerRank.

1 - Part 1: B-Tree Design (40 pts.)

In this homework, we will work on B-trees. We have a simple node structure having the following three attributes:

- **int** x
- **int** y
- **char** z

Create a node structure to store these attributes in addition to other pointer attributes to create the tree. Then, create the tree according to the given inputs. Here the first input indicates the total node count, second input indicates degree of the tree, the third input indicates which attribute should be used as the key. The other inputs are to assign x,y and z values respectively. An example input is given below.

```
21
3
z
56 34 G
71 6 M
68 0 P
123 -666 T
999 4 X
41 33 A
-66 8 B
748 54 C
99 978 D
400 23 E
98 66 J
0 43 K
66 12 N
45 1 O
11 -34 Q
67 -36 R
40 7 S
85 3 U
8 2 V
62 9 Y
9 5 Z
```

The output for the input above should be a traverse of the tree in prefix order. Every node should be written in a different line.

```
(748,54,C) (56,34,G) (68,0,P) (40,7,S) (8,2,V)
(41,33,A) (-66,8,B)
(99,978,D) (400,23,E)
(98,66,J) (0,43,K) (71,6,M) (66,12,N) (45,1,O)
(11,-34,Q) (67,-36,R)
(123,-666,T) (85,3,U)
(999,4,X) (62,9,Y) (9,5,Z)
```

Compare your outputs with the outputs from HackerRank. For full grades, you should obtain true outputs for each of the cases.

2 - Part 2: Delete Operation (40 pts.)

In this part, another line is added to the input file containing information of which key should be deleted from the tree. After creating the tree, delete the given key from the tree and print the tree as you have done in the previous part.

3 - Part 3: ”What if...” (20 pts.)

Using at most 500 characters in a txt file, answer the following questions.

- Suppose that, using the same structure, we created three different bonds according to different x,y, z values. Thus, for every node we have the information of *x_children*, *y_children*, *z_children* and *x_parent*, *y_parent*, *z_parent* etc. What is the complexity of insertion operation for this structure?

- Suppose that, instead of storing x,y,z values in a node, our B-tree has a different node structure containing mini B-Trees as given in Figure 1. The key of each node for the tree is calculated as standard deviation of x values inside the mini B-tree. What is the complexity of adding a new node to one of the mini B-trees?

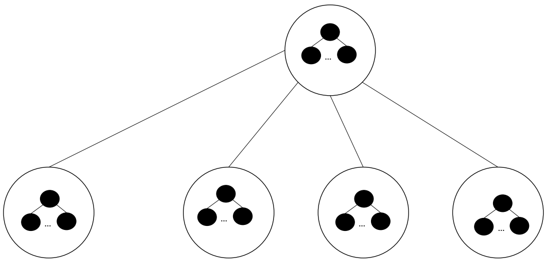

Figure 1: Mini B-Trees inside another B-tree