CS225 Assignment 8

Group 18: Xu Ke, Wang Dingkun, Qian shuaicun, Hua Bingshen

April 2021

1 Assignment 8 Exercise 1

1.1

Find the minimum key stored in a B-tree: it is clear that for key stored in it satisfies the condition that the one in the left is less than the one in the right (at least in my code), thus to find the minimum key is to find the leftmost leaf. Thus, we only need to traverse from root recursively to the leaf level and find the first one with index 0 in my vector.

Find the maximum key stored in a B-tree: similarly, it is clear that for key stored in it satisfies the condition that the one in the right is larger than the one in the left (at least in my code), thus to find the maximum key is to find the rightmost leaf. Thus, we only need to traverse from root recursively to the leaf level and find the last one and here we can use size() function to get the length of my vector.

1.2

Please see the code in the file attached.

1.3

Find the predecessor key of a given key stored in a B-tree:

There should be three cases shown as following:

Case 1: Leaf, predecessor in Node (index > 0)

Case 2: Nonleaf, predecessor in Child (index == 0)

Case 3: Leaf, predecessor in Parents (the last one less than key, or the maximum in the parents node)

Find the successor key of a given key stored in a B-tree:

There should be three cases shown as following:

Case 1: Leaf, successor in Node (index < size-1)

Case 2: Nonleaf, successor in Child (index == size-1)

Case 3: Leaf, successor in Parents (the first one larger than key, or the minimum in the parents node)

1.4

Please see the code in the file attached.

2 Assignment 8 Exercise 2

2.1

Find the minimum key stored in a B+tree: it is clear that for key stored in it satisfies the condition that the one in the left is less than the one in the right (at least in my code), thus to find the minimum key is to find the leftmost leaf. Thus, we only need to traverse from root recursively to the leaf level and find the first one with index 0 in my vector, which is what we can also apply from B-tree. Of course, due to the special property of B+tree that it only stores data in the leaf with doubly-linked list, thus what we only need to do is to go depth to the leaf level and find the first element with index 0.

Find the maximum key stored in a B+tree: similarly, it is clear that for key stored in it satisfies the condition that the one in the right is larger than the one in the left (at least in my code), thus to find the maximum key is to find the rightmost leaf. Thus, we only need to traverse from root recursively to the leaf level and find the last one and here we can use size() function to get the length of my vector, which is what we can also apply from B-tree. Of course, due to the special property of B+tree that it only stores data in the leaf with doubly-linked list, thus what we only need to do is to go depth to the leaf level and find the last element with the help of size() function.

2.2

Please see the code in the file attached.

2.3

Similarly, we can also apply what we have for B-tree with small modification here shown as following:

Find the predecessor key of a given key stored in a B+tree: There should be two cases shown as following: **Case 1:** Leaf, predecessor in Node (index > 0) **Case 2:** Leaf, predecessor in the last Child of the previous one element in Parents. **Find the successor key of a given key stored in a B+tree:** There should be two cases shown as following: **Case 1:** Leaf, successor in Node (index < size-1) **Case 2:** Leaf, successor in the first Child of the previous one element in Parents.

In addition, due to the special property of B+tree that it only stores data in the leaf with doubly-linked list, thus what we only need to do is to go depth to the leaf level and find the location of the interested key with index i, and then the predecessor is just the key before with index i-1 and the successor is just the key after with index i+1.

2.4

Please see the code in the file attached.

3 Assignment 8 Exercise 3

Please see the code in the file attached.

4 Assignment 8 Exercise 4

4.1

Please see the code in the file attached.

4.2

Please see the code in the file attached.