Ex1

(i)

To find the minimum key stored in a B-tree, we start searching from the root node, then we always choose to move to the first child of the parent node until we reach a leaf node. Since keys in each node are sorted in a total order, the first key in this leaf node must be the minimum key in the B-tree. The same procedure applies to finding the maximum key. Instead of going to the first child, we choose to move to the last child until reaching a leaf node. Then the key with highest index in this leaf node is the maximum key in the B-tree.

(iii)

We will assume that every node in the B-tree has a pointer to its parent node. We also assume that the key has index [a] in the node.

Predecessor

1. If the given key is in a non-leaf node:
Move to the left child (with index [a] in child list) of this key, then always choose to move to the last child of the parent node until we reach a leaf node. Then the key with highest index in this leaf node is the predecessor of the given key.
2. If the given key is in a leaf node but not the first element in key list:
Simply return the key with index [a-1], this must be its predecessor.
3. If the given key is the first element in key list of a leaf node:
First move to its parent node and check if this leaf node is the first leaf node of its parent:
a. If not, then the key with index [a-1] in its parent node is the predecessor.
b. If so, then move to the parent node of this parent node recursively until the current node is not the first child of its parent. Then the key with index [a-1] in its parent node is the predecessor. If we reach the root node and it is still the first child, then the key is the smallest key in the tree and dose not have predecessor.

Successor

1. If the given key is in a non-leaf node:
Move to the right child (with index [a+1] in child list) of this key, then always choose to move to the first child of the parent node until we reach a leaf node. Then the key with smallest index in this leaf node is the successor of the given key.
2. If the given key is in a leaf node but not the last element in key list:
Simply return the key with index [a+1], this must be its successor.
3. If the given key is the last element in key list of a leaf node:
First move to its parent node and check if this leaf node is the last leaf node of its parent:

a. If not, then the key with index [a+1] in its parent node is the successor.
b. If so, then move to the parent node of this parent node recursively until the current node is not the last child of its parent. Then the key with index [a+1] in its parent node is the successor. If we reach the root node and it is still the last child, then the key is the largest key in the tree and dose not have successor.