

University of Florida

---

Project Report to COP 5536

---

Fall 2017

## ADS Project Report



Submitted By:  
Aroushi Sharma  
UFID: 5004-5862

[aroushisharma@ufl.edu](mailto:aroushisharma@ufl.edu)

# 1 Problem Statement

Develop and test a class B+Tree to store pairs of the form (key, value). For this project you are to implement a memory resident B+tree (i.e., the entire tree resides in main memory). Your implementation must be able to store multiple pairs that have the same key (i.e., duplicates). The leaves should be linked into a doubly linked list to support efficient range retrieval. The supported operations are:

1. Initialize(m): create a new order m B+Tree
2. Insert (key, value)
3. Search (key): returns all values associated with the key
4. Search (key1, key2): returns (all key value pairs) such that  $\text{key1} \leq \text{key} \leq \text{key2}$ .

## 2 Solution Description

To implement the B-Plus Tree structure having the Key Value pair, Java was used as the programming Language. The code structure was divided into six classes whose description and function prototypes are given below:

### **Class BPlusTrees**

This class contained all the functions that the B-Plus tree is required to perform. These include the function declaration of insert(Key, value), search(key), and search(key1,key2). Using this Class as the base different classes were called based on normal search and range search. These search criteria are different based on what we try to retrieve.

Function Prototypes:

```
void insert(double,String); //inserts the Key and the value to the tree
```

```
ArrayList<String> search(double); //searches for the value associated with the key given as parameter
```

```
ArrayList<KeyValPair> search(double, double); //searches for all the values associated with the given key Range
```

### **Class KeyValPair**

This class is used to store the key-value pair required when doing range search and get or set its value.

Function Prototypes:

```
double getPairKey(); //gets the key in a Key-Value pair
```

```
double getPairValue(); //gets the value associated with the key-Value pair
```

```
String toString(); //converts the output to the desired format
```

### **Abstract Class BPlusNode**

This is an abstract Class whose implementation are provided by BPlusSearch and BPlusRangeSearch. It also has abstract functions for the insert function into the B-Plus Tree and splits and merge function when the tree becomes overfull.

Function Prototypes:

boolean isOverfull(); //checks if the tree is above the given order m

boolean isPair(); //checks if we are looking for a Key-Value pair

void Insert(KeyValPair); //abstract function

BPlusNode searchSubNode(double); // abstract function

BPlusNode searchSubNode(KeyValPair); // abstract function

void search(double,double,ArrayList<KeyValPair>); // abstract function

BPlusSearch split(); //abstract function

void merge(BPlusSearch); //abstract function

### **Class BPlusSearch (implemets BPlusNode)**

This class is used to implement searching the corresponding Value of the given key. This class implements the Abstract Class BPlusNode.

Function Prototypes:

ArrayList<Double> getKeys() //To get all the keys

ArrayList<BPlusNode> getSubNodes() / to get all the sub nodes

int searchIndex(double); //to get the index by using binary search

void addFirst(BPlusNode); //to add the root node

BPlusNode searchSubNode(double); //search function to get the subnodes of value associated with the key

BPlusNode searchSubNode(KeyValPair); //search function to get the subnodes of key-value associated with the key range

BPlusSearch split(); //To split the node if it is overfull

void merge(BPlusSearch); //Merger node

String toString(); //to convert to given output format

### **Class BPlusRSearch (implements BPlusNode)**

This class is used to implement searching the corresponding Key-Value Pairs of the given key range. This class implements the Abstract Class BPlusNode.

Function Prototypes:

```
int searchIndex(double); //To search the index using Binary search
void insert(KeyValPair); //inserts a Key-Value pair into the tree
void search(double,ArrayList<String>) //Search to match with the given key
void search(double,double,ArrayList<KeyValPair>) // Search to match with the given key
BPlusSearch split(); //Split an overfull node
String toString(); //Convert to the required output format
```

### **Class treesearch**

This class is the entry point in the program which takes the input file as argument and writes the result to the output\_file.txt.

```
public static void main(String[])
```