Name: Anirudh Pathak
UFID: 64091067
UF Email account: paanir@ufl.edu

Classes and Methods

treesearch

The 'main' class that reads input from the input file, parses it and calls appropriate functions

Tree

Denotes the B+Tree

Public functions:

insert

```
void insert(double key, String value)
```

inserts a key-value in the B+Tree

search(key)

```
List<String> search(double key)
```

Searches for value of the passed key

search(start, finish)

```
List<BEntry> search(double start, double finish)
```

search for all key-value pairs between start and finish

Private functions:

merge

```
void merge(TreeNode treeIdxNode, TreeNode loneIdxNode)
```

assists insert() for bubbling up overfull nodes

TreeNode

Fields:
isDataNode: it is index node or data node
parentNode: pointer to its parent node
indices: index list for index nodes
children: list of children for index nodes
dataList: list of BEntries for data nodes

BEntry

An entry in the B+Tree
It has the following fields:

key: Key of the entry
values: List of values for the key
prev and next pointers: For the doubly linked list between all other BEntries

BPair

Key-value pairs for input to the B+Tree

StringUtils

Functions required to parse the input

```
boolean isInsert(String s)
```

checks if the input string is an insert

```
BPair getInsertPair(String s)
```

Extract key and value from insert line

```
boolean isSearch(String s)
```

checks if the input string is a search

```
double getSearchKey(String s)
```

Extract key from search string

```
boolean isSearchRange(String s)
```

checks if the input string is a search range

```
double[] getSearchRange(String s)
```

extract search range from search range string

TreeUtils

Utility functions required by the Tree class:
searchIdxList:

```
int searchIdxList(List<Double> indices, double key)
```

searches for a key in the Index list of an index node

searchDataList:

```
int searchDataList(List<BEntry> dataList, double key)
```

searches for a key in the data list of a data node

insertInDataNode:

```
void insertInDataNode(TreeNode dataNode, double key, String value)
```

inserts a key-value pair inside a datanode. Also, adds it to the linked list (initializes prev and next pointers)

searchForDataNode:

```
TreeNode searchForDataNode(double key, TreeNode currNode
```

searches for the right data node in the B+tree

createIndexNode:

```
TreeNode createIdxNode(TreeNode parentNode, List<Double> indices, List<TreeNode> children)
```

creates an index node

createDataNode:

```
TreeNode createDataNode(TreeNode parentNode, List<BEntry> dataList)
```

creates a data node