**Source Code**

**B+ Tree**

**S. Cloud State University**

**Team 12**

**Adcock Hunter, hm3101dq**

**Kushal Singh, vu4055bm**

**Sabin Basnet, bp4065pg**

**Ashish Dahal, xc0728qf**

**Mohammad Hejazi Najdi, xb8321ui**

**Header file or .h file**

**#pragma once**

**#include <string>**

**#include <sstream>**

**#include <vector>**

**using namespace std;**

**struct kp {**

**string key;**

**int value;**

**kp(string nkey,int value) {**

**this->key = nkey;**

**this->value = value;**

**}**

**void setValue(int nValue) {**

**this->value = nValue;**

**}**

**int getValue() {**

**return this->value;**

**}**

**};**

**struct sequenceB {**

**int nextPointer;**

**int prevPointer;**

**vector<string> seqeunce;**

**sequenceB() {**

**}**

**sequenceB(int prev,string item) {**

**this->prevPointer = prev;**

**this->seqeunce.push\_back(item);**

**}**

**void setNext(int next){**

**this->nextPointer = next;**

**}**

**string writeToFile() {**

**string temp;**

**stringstream ss(temp);**

**temp.push\_back('s');**

**//temp.push\_back('|');**

**//temp.push\_back(prevPointer);**

**temp.push\_back('|');**

**for (int i = 0; i < seqeunce.size(); i++) {**

**for (int x = 0; x < seqeunce.at(i).size(); x++) {**

**temp.push\_back(seqeunce.at(i).at(x));**

**}**

**//ss << seqeunce.at(i);**

**//cout << "What is going into the sequence " << seqeunce.at(i) << endl;**

**//ss >> temp;**

**}**

**//cout << "THE LAST ITEM IN THE SEQEUENCE " << seqeunce.at(2)<<endl;**

**//cout << "THE SIZE OF THE SEQUENCE BLOCK " << seqeunce.size() << endl;**

**temp.push\_back('|');**

**temp.push\_back(nextPointer);**

**temp.push\_back('|');**

**//cout << "WHAT IS BEING RETURNED TO BE PRINTED " << temp << endl;**

**//for (int i = 0; i < seqeunce.size(); i++) {**

**// cout << "IN THE SEQUENCE " << seqeunce.at(i) << endl;**

**//}**

**return temp;**

**}**

**};**

**struct indexB {**

**int nextPointer;**

**int prevPointer;**

**vector<kp> seqeunce;**

**indexB() {**

**}**

**indexB(int prev, kp item) {**

**this->prevPointer = prev;**

**this->seqeunce.push\_back(item);**

**}**

**void setNext(int next) {**

**this->nextPointer = next;**

**}**

**string writeToFile() {**

**string temp;**

**stringstream ss(temp);**

**temp.push\_back('i');**

**temp.push\_back('|');**

**for (int i = 0; i < seqeunce.size(); i++) {**

**ss >> temp;**

**}**

**temp.push\_back('|');**

**return temp;**

**}**

**};**

**class bplustreemk4 {**

**private:**

**//Header Record Variables**

**int headerSize;**

**int insertFlag;**

**int blockSize ;**

**int indexBlockSize;**

**int totalSequnceBlocks;**

**int totalIndexBlock;**

**int numAvalBlocks;//Number of open blocks**

**int numOfLevels;**

**string header;**

**int rbnRoot; //Relative offest to root node**

**int rbnLevels;//Relative offset to levels**

**int rbnAvailList;//Relative offset to the avial list**

**int totalNumRecords;//total number of records**

**int sizeofRecords;**

**int fieldsPerRecord;**

**vector<string> sizePerfield;**

**vector<char> labelPerfield;**

**int sortField;**

**string description;**

**string time;**

**string lastMod;**

**string dir;**

**public:**

**bplustreemk4();**

**bplustreemk4(string directory);**

**void processFile();**

**void cleanUp();**

**};**

**Implementation file or .cpp file**

**#include <iostream>**

**#include <fstream>**

**#include <string>**

**#include <sstream>**

**#include <vector>**

**#include "bplustreemk4.h"**

**using namespace std;**

**bplustreemk4::bplustreemk4(string dir) {**

**//ofstream out;**

**fstream myfile;**

**this->dir = dir;**

**string header;**

**string headerout;**

**myfile.open(dir, ios::in);**

**myfile >> this->headerSize;**

**getline(myfile, header);**

**cout << header << endl;**

**this->header = header;**

**int fieldcount = 0;**

**int count = 0;**

**//a "different" way to read data in because it wasnt working**

**for (int i = 1; i < this->headerSize - 2; i++) {**

**if (header.at(i) == '|') {**

**cout << "encounter a field marker" << endl;**

**fieldcount++;**

**string temp;**

**for (int x = i - count; x < i; x++) {**

**temp.push\_back(header.at(x));**

**}**

**if (fieldcount == 1) {**

**this->insertFlag = stoi(temp);**

**}**

**if (fieldcount == 2) {**

**this->totalNumRecords = stoi(temp);**

**}**

**if (fieldcount == 3) {**

**this->sizeofRecords = stoi(temp)-1;**

**}**

**if (fieldcount == 4) {**

**this->fieldsPerRecord = stoi(temp);**

**}**

**if (fieldcount == 5) {**

**cout << "temp aka what ever we have deemed a field " << temp;**

**/\*vector<string> temp2;**

**for (int y = 0; y < temp.size(); y++) {**

**}\*/**

**vector<string> temp2;**

**string buffer;**

**stringstream ss(buffer);**

**for (int y = 0; y < temp.size(); y++) {**

**ss.clear();**

**ss << temp.at(y);**

**//cout << "Went into the string stream" << temp.at(y) << endl;**

**ss >> buffer;**

**//cout << "THIS IS WHAT IS IN THE BUFFER " << buffer << endl;**

**if (buffer != "\"" || buffer != "(" || buffer != ")") {**

**temp2.push\_back(buffer);**

**}**

**}**

**this->sizePerfield = temp2;**

**}**

**if (fieldcount == 6) {**

**vector<char> temp3;**

**for (int y = 0; y < temp.size(); y++) {**

**if (temp.at(y) != '"'|| temp.at(y)!='(' || temp.at(y) != ')') {**

**temp3.push\_back(temp.at(y));**

**}**

**}**

**this->labelPerfield = temp3;**

**}**

**if (fieldcount == 7) {**

**this->sortField = stoi(temp);**

**}**

**if (fieldcount == 8) {**

**this->description = temp;**

**}**

**count = 0;**

**i++;**

**}**

**count++;**

**}**

**myfile.close();**

**this->rbnRoot = this->header.size();**

**//debugging**

**cout << this->headerSize << endl;**

**cout << this->totalNumRecords << endl;**

**cout << this->description << endl;**

**cleanUp();**

**processFile();**

**}**

**bplustreemk4::bplustreemk4() {**

**}**

**void bplustreemk4::processFile() {**

**ofstream outfile;**

**fstream myfile;**

**myfile.open(this->dir);**

**outfile.open("btree.dat");**

**outfile << this->header;**

**sequenceB sb;**

**indexB ib;**

**//pair temp;**

**int nextP=0;**

**int preP=0;**

**this->totalSequnceBlocks = 0;**

**string tempLine;**

**while (getline(myfile, tempLine)) {**

**string zip;**

**if (tempLine.size() != this->sizeofRecords) {**

**cout << "Line that was not big enough dected." << endl;**

**continue;**

**}**

**else {**

**cout << "The line is of valid lenghth " << endl;**

**zip = tempLine.substr(0, 5);**

**cout << " The zip is " << zip << endl;**

**sb.nextPointer = this->headerSize + (this->blockSize \* this->totalSequnceBlocks);**

**cout << "Next pointer is " << sb.nextPointer << endl;**

**cout << "pushing back this : " << tempLine << endl;**

**sb.seqeunce.push\_back(tempLine);**

**cout << " SEQUENCE SIZE BEFORE ADDING " << sb.seqeunce.size() << endl;**

**ib.seqeunce.push\_back(kp(zip, (this->headerSize + this->totalSequnceBlocks \* 249)));**

**if (sb.seqeunce.size() == 3) {**

**this->totalSequnceBlocks++;**

**//cout << "this is what is in the sequence block " << sb.writeToFile() << endl;**

**outfile << sb.writeToFile();**

**sb.seqeunce.clear();**

**continue;**

**}**

**if (ib.seqeunce.size() == this->indexBlockSize-1 ) {**

**this->totalIndexBlock++;**

**cout << "this is what is in the Index block " << ib.writeToFile() << endl;**

**outfile << ib.writeToFile();**

**ib.seqeunce.clear();**

**continue;**

**}**

**//cout << "The sequence size after adding " << sb.seqeunce.size() << endl;**

**}**

**}**

**outfile.close();**

**myfile.close();**

**cout << this->sizeofRecords << endl;**

**}**

**void bplustreemk4::cleanUp() {**

**vector<string> tempf;**

**string temp;**

**for (int i = 0; i < this->sizePerfield.size(); i++) {**

**if (this->sizePerfield.at(i) != "\"" && this->sizePerfield.at(i) != "(" && this->sizePerfield.at(i) != ")") {**

**cout << this->sizePerfield.at(i);**

**tempf.push\_back(this->sizePerfield.at(i));**

**}**

**}**

**this->sizePerfield = tempf;**

**cout << endl;**

**for (int i = 0; i < this->labelPerfield.size(); i++) {**

**cout << this->labelPerfield.at(i);**

**}**

**}**

**Main file/ main.cpp**

**#include <string>**

**#include <iostream>**

**#include <sstream>**

**#include <fstream>**

**#include "bplustreemk4.h"**

**#include "bplustreemk4.cpp"**

**using namespace std;**

**string dir = "C:\\Users\\reali\\Desktop\\CSCI\\us\_postal\_codesn+.txt";**

**//string dir = "C:\\Users\\reali\\Downloads\\drive-download-20181213T164829Z-001\\us\_postal\_codesn+.txt";**

**int main() {**

**/\*Due to the lack of time this is implemented staticly\*/**

**bplustreemk4 test = bplustreemk4(dir);**

**return 0;**

**}**