**Test Document**

**B+ Tree**

**S. Cloud State University**

**Team 12**

**Adcock Hunter, hm3101dq**

**Kushal Singh, vu4055bm**

**Sabin Basnet, bp4065pg**

**Ashish Dahal, xc0728qf**

**Mohammad Hejazi Najdi, xb8321ui**

**Description**

For are testing we used the provided U.S Postal codes. We did this because after learning on Wednesday that we had to completely rewrite our implementation of the B+ tree. In the time between Wednesday and Friday we have gotten it almost work 100% the right way but statically.

Some problems that we encountered along the way where the fact that when trying to grab single letters are out of strings you end up getting stuck in this converting back and forth loop.

Another problem encounter was how to do all the relative byte offsets to the different sequence blocks when you had index blocks mixed in. This is part of the program that does not function 100%. One thing we would like to get in was a better menu. But due to the fact we had to start from scratch on Wednesday we couldn't fit it in. We also would liked to make a nice way to print the tree out but due to the very bad circumstance of having to rewrite the entire program the correct way we ran out of time to add this nice visual touch to the program. Another nifty thing we would've liked to get implemented on time was the able to sort by a different field. This is a slightly complex task because reading in list from a file is a pain in the butt because of char conversion to string and then string to int because we could not find a good way to convert straight from char to int.

One bug we ran into that was really crippling was when piping in data from the file the typical “myfile >> var1 >>var2 >>var3” failed after the second one even with proper delimiters and everything. So to get data from a file and assign it to variables we had to go to really annoying lengths to circumvent this.

**After the compilation of the program, following are the some of the test conditions.**

**Test run for Larger file**

Next pointer is 41024

pushing back this : 99926Metlakatla AK Prince of Wales-Hyder 55.1215 -131.579

SEQUENCE SIZE BEFORE ADDING 3

The line is of valid lenghth

The zip is 99927

Next pointer is 41027

pushing back this : 99927Point Baker AK Prince of Wales-Hyder 56.3528-133.6211

SEQUENCE SIZE BEFORE ADDING 1

The line is of valid lenghth

The zip is 99928

Next pointer is 41027

pushing back this : 99928Ward Cove AK Ketchikan Gateway 55.3954-131.6754

SEQUENCE SIZE BEFORE ADDING 2

The line is of valid lenghth

The zip is 99929

Next pointer is 41027

pushing back this : 99929Wrangell AK Wrangell-Petersburg (C 56.4335-132.3529

SEQUENCE SIZE BEFORE ADDING 3

The line is of valid lenghth

The zip is 99950

Next pointer is 41030

pushing back this : 99950Ketchikan AK Ketchikan Gateway 55.3422-131.6478

SEQUENCE SIZE BEFORE ADDING 1

81

**Test run for shorter file:**

The zip is 2675

Next pointer is 737

pushing back this : 2675Yarmouth Port MA Barnstable 41.7051 -70.227

SEQUENCE SIZE BEFORE ADDING 1

The line is of valid lenghth

The zip is 2702

Next pointer is 737

pushing back this : 2702Assonet MA Bristol 41.7975 -71.0607

SEQUENCE SIZE BEFORE ADDING 2

The line is of valid lenghth

The zip is 2703

Next pointer is 737

pushing back this : 2703Attleboro MA Bristol 41.9296 -71.3009

SEQUENCE SIZE BEFORE ADDING 3

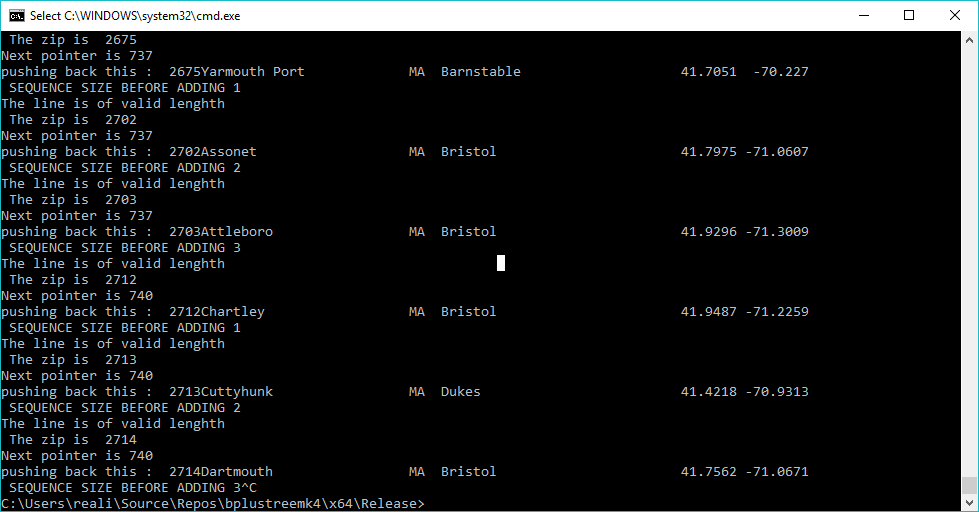
The line is of valid lenghth

The zip is 2712

Next pointer is 740

pushing back this : 2712Chartley MA Bristol 41.9487 -71.2259

this is what we got after running the program



Script file is included in separate file which is also the part of test document.