

# CS 202, Summer 2013

## Homework Assignment 2

Due: 23.59, July 16, 2013

In this homework assignment, you are supposed to implement a binary search tree and 2-3 tree with simple insert and retrieve methods. Your program will read the contents from an input file which includes a list of English words, build corresponding binary search tree and 2-3 tree, and then retrieve the words given in a query file.

For example, corresponding BST and 2-3 tree is given below for a sample input file. Each line represents a word in input file. You need to test your program with the exact input file format. You can find large input files by searching internet:

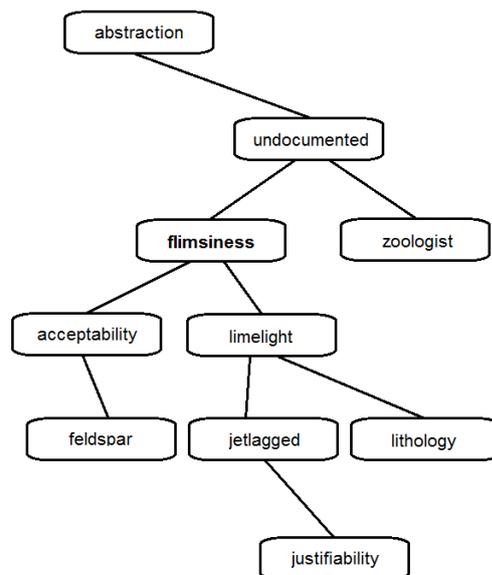
(i.e. [http://www.cs.bilkent.edu.tr/~arpa/corncob\\_lowercase.txt](http://www.cs.bilkent.edu.tr/~arpa/corncob_lowercase.txt))

Assume that all words are unique. String comparisons should be case insensitive (i.e., the words "abstraction" and "Abstraction" should have the same key values.).

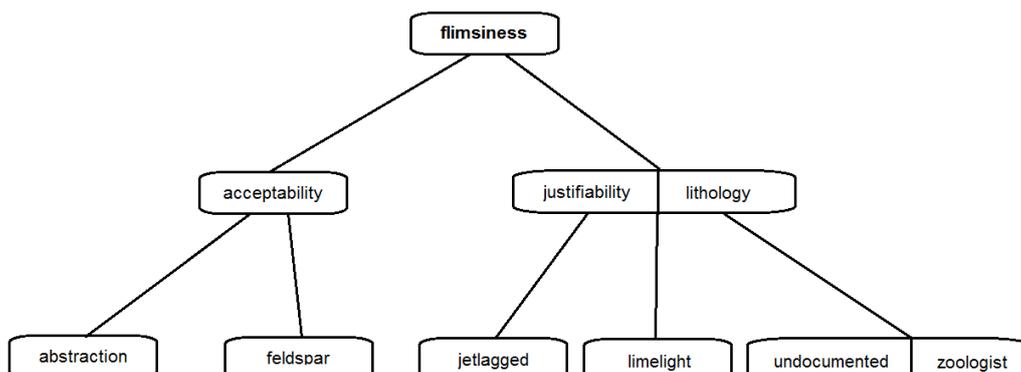
### Sample Input File (input.txt)

```
abstraction
undocumented
flimsiness
acceptability
limelight
lithology
feldspar
jetlagged
justifiability
zoologist
```

### Binary Search Tree



### 2-3 Tree



After building the trees, you need to retrieve the words given in a query file and print the results into an output file. A sample query file and output file are given below. Similar to input file, each line represents a word in query file. Your program will search each word in both BST and 2-3 tree and then print the level of node for each tree in the output file. For instance, by considering input file above, your program should print "1 3 abstraction" for the word "abstraction". This output means that the level of node that abstraction is placed is 1 for BST, and 3 for 2-3 Tree. If the word is not found, print 0 for the level of the word. Test your program with the exact file formats given below.

#### Sample Query File (query.txt)

```
soprano
acceptability
feldspar
spectator
justifiability
zoologist
abstraction
limelight
```

#### Sample Output File (output.txt)

```
0 0 soprano
4 2 acceptability
5 3 feldspar
0 0 spectator
6 2 justifiability
3 3 zoologist
1 3 abstraction
4 3 limelight
```

The name of the input file, query file and output file will be provided as command line arguments to your program. Thus, we call your programs using two command line arguments in the following format:

```
username@dijkstra:~>./levelfinder <inputFileName> <queryFileName> <outputFileName>
```

#### Notes:

1. This assignment is due by 23.59 on Tuesday, July 16th, 2012. You should upload your homework using Moodle course page. No hardcopy submission is needed. The standard rules about late homework submissions apply. Please see the course syllabus for further discussion of the late homework policy as well as academic integrity.
2. Your code must not have any memory leaks. You will lose points if you have memory leaks in your program even though the outputs of the operations are correct.
3. You should upload all files as a single archive file (i.e. zip file). The name of archive file should be in following format; *id\_surname\_name.zip*. The submissions that do not obey these rules will not be graded.
4. You are free to write your programs in any environment (you may use either Linux or Windows). On the other hand, we will test your programs on "dijkstra.ug.bcc.bilkent.edu.tr" and we will expect your programs to compile and run on the "dijkstra" machine. If we could not get your program properly work on the "dijkstra" machine, you would lose a considerable amount of points. Therefore, we recommend you to make sure that your program compiles and properly works on "dijkstra.ug.bcc.bilkent.edu.tr" before submitting your assignment.
5. This homework will be graded by your TA Sami Arpa (arpa@cs.bilkent.edu.tr). Thus, you may ask your homework related questions directly to him.