ITU Computer Engineering Department

BLG 223E Data Structures, Fall 2021-2022

Recitation #5

Due Date: 22.12.2021, 23:59

You are asked to construct a trie to keep the list of BLG223E students using their student numbers. The trie will be used to search for an enrolled student with his/her student number.

Input: The list of student numbers that enrolled in the BLG223E class are stored in a text file whose name will be given as a **command line argument** (e.g., BLG223E_students.txt). The file contains exactly 10 student numbers per line (except the last line that contains at least 1 and at most 10 student numbers) which are separated by a space character, as shown in Figure 1. All the student numbers consist of 9 digits.

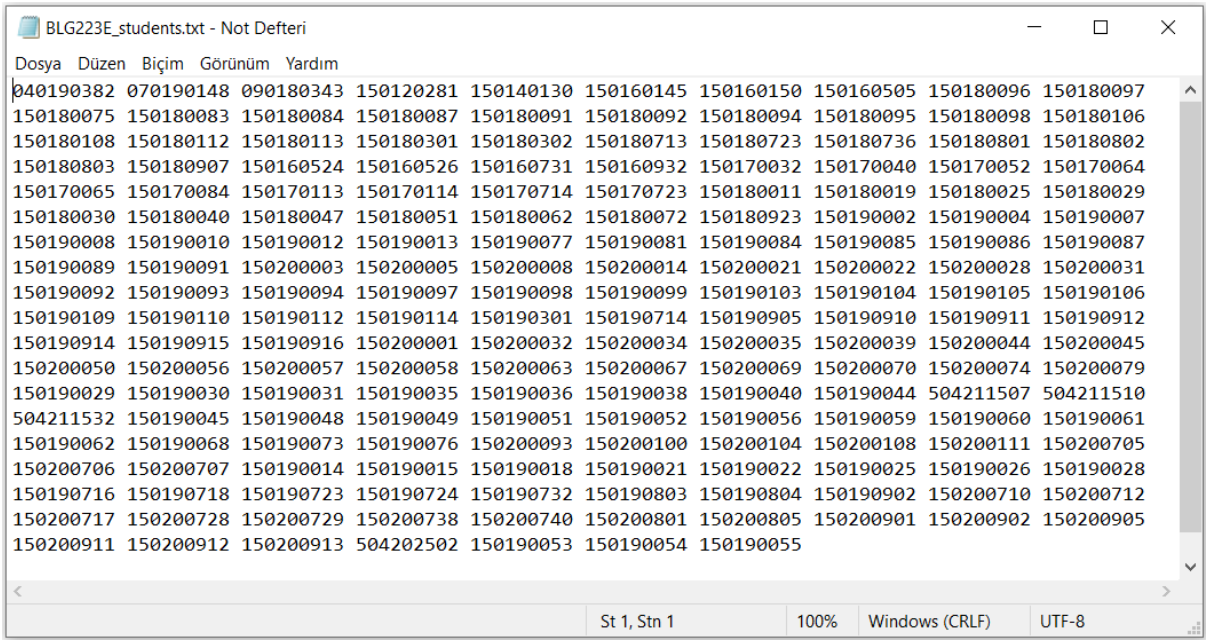

Figure 1: Student numbers of the students are kept in a text file

Based on our example BLG223E_students.txt file, Figure 2 shows the partly constructed trie when the student numbers in the first line of the BLG223E_students.txt file are inserted into the trie. When the trie construction is fully completed, the trie will contain nodes for all the student numbers in the BLG223E_students.txt file.

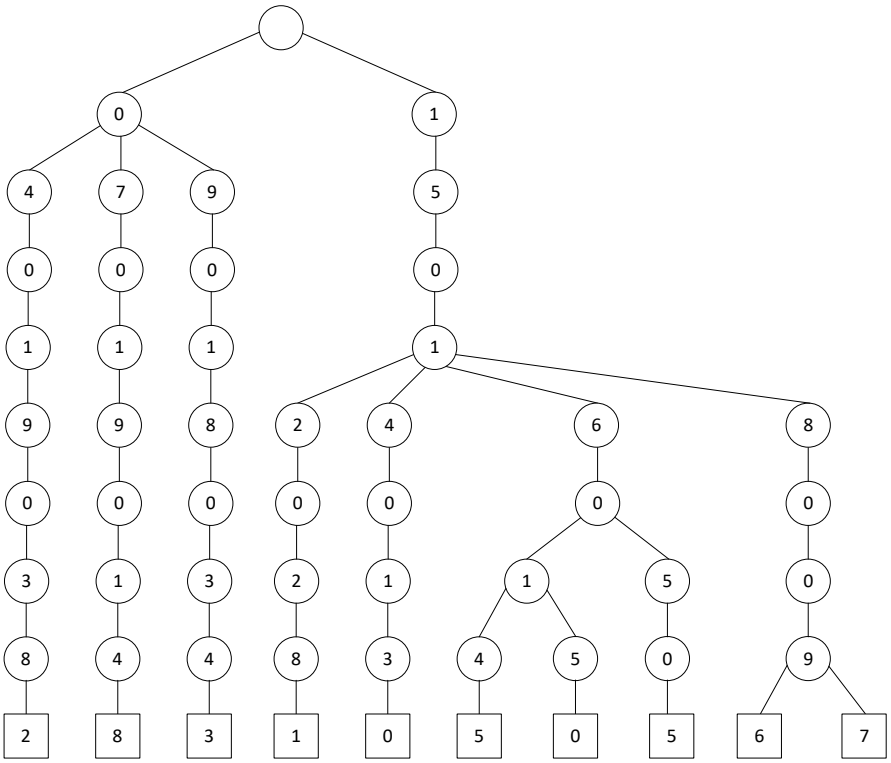

Figure 2: Trie after processing numbers in the first line of example BLG223E_students.txt text file

Wanted: You are given the following C++ StudentsTrie.h header file that defines StudentsTrie data structure along with TrieNode (Figure 3).

```
// PLEASE DO NOT REMOVE DECLARED VARIBLES AND FUNCTIONS OF CLASSES,
// IF YOU PREFER, YOU CAN ADD NEW ONES
#ifndef _H
#define _H

using namespace std;

#define MAX_CHILDREN 10 // Each student ID consists of a sequence of digits: 0-9

class TrieNode{
        char digit;   // Current Digit
        TrieNode* children[MAX_CHILDREN];   // Next Digit(s)
        friend class StudentsTrie;

    public:
        TrieNode(const char& digit);
};

class StudentsTrie{
        TrieNode *root;

    public:
        StudentsTrie     ( const string& file_name  ); // Construct a StudentsTrie using the records in 'file_name'
        void insert_id   ( const string& student_id ); // Insert a student ID into the StudentsTrie
        bool in_the_trie ( const string& student_id ); // Check existence of a student ID in the StudentsTrie
        ~StudentsTrie    (                          ); // StudentsTrie Destructor
};

#endif
```

Figure 3: StudentsTrie.h Header File for StudentsTrie data structure

You must implement the methods given in the StudentsTrie.h header file in a StudentsTrie.cpp file so that the following main.cpp file (Figure 4) works as described below.

```
1   /*
2   PLEASE DO NOT CHANGE THIS FILE, OTHERWISE YOUR CODE WILL NOT BE EVALUATED AND WILL BE GRADED AS 0
3   */
4
5   #include <iostream>
6
7   #include "StudentTrie.h"
8
9   using namespace std;
10
11  int main(int argc, char* argv[]){
12      //system("clear");// make this line as comment if you are compiling on Windows
13      //system("cls"); // make this line as comment if you are compiling on Linux or Mac
14
15      StudentsTrie st(argv[1]);
16
17      cout << "is " << argv[2] << " in the trie: " << boolalpha << st.in_the_trie(argv[2]) << endl;
18
19      return EXIT_SUCCESS;
20  }
```

Figure 4: main.cpp File

The pseudocode for the StudentsTrie constructor (Line 15) is given below:

(Step 1) Get the **file_name** as an actual parameter (in our case, **argv[1]**).

(Step 2) Open the **file_name** in read mode.

(Step 3) Read a **student_number** from the file.

(Step 4) Insert **student_number** to the trie using **insert_id()** method of the StudentsTrie class where **student_number** is the last student number read in **(Step 3)**.

(Step 5) If there are more student numbers in the **file_name**, continue with **(Step 3)** to read next **student_number**. Otherwise, continue with **(Step 6)**.

(Step 6) Close the **file_name** and finish constructor processing.

After constructing the StudentsTrie data structure, you must search for a student number which is given as the **second command line argument (i.e., argv[2])**. The **in_the_trie()** method of StudentsTrie class is used to search for a student ID within the StudentsTrie (**Line 17**). If the searched number is in the StudentsTrie, **in_the_trie()** method returns true. Otherwise, **in_the_trie()** method returns false.