**NLP Assignment-2 Report**

**Ibrahim Ismail Erhan 20155510026**

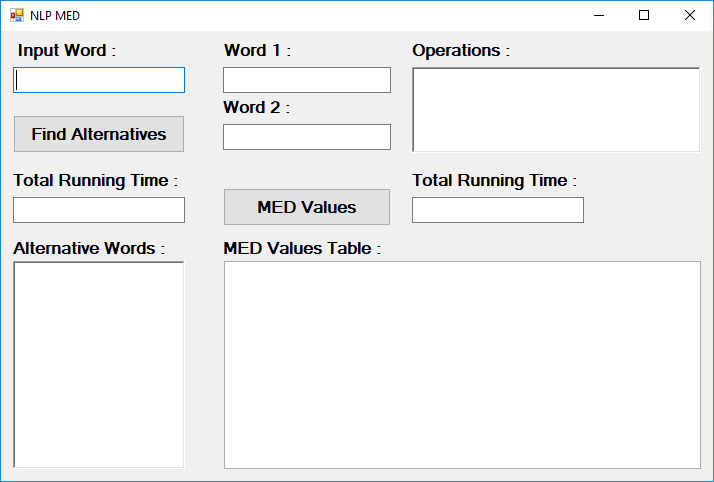
1. **Definitions of Algorithms**

* **class Word** implemented to keep words from dictionary file(“sozluk.txt”). Attribute “str” keeps word data and “cost” keeps cost value relative to the input word taken from user.
* **class MED** includes all algorithms asked to implement in assignment document. The function ant procedures are explained blow;
  + **public MED(string dictionaryFilePath)** 🡪 is the constructer of the class takes dictionary file path, reads it and initiates “wordDictionary” list.
  + **private List<Word> wordDictionary**;🡪 is a “Word” list keeps all world read from dictionary file.
  + **public List<Word> FindAlternates(string word)**🡪 The function that takes a word and return alternative similar 5 word according to it.
  + **static public int EditDist(string str1, string str2, int m, int n)** 🡪 This is a recursive utility function intended to use in FindAlternate function. Takes two word and calculates minimum edit distance between these words. (“m” and “n” parameters are lengths of str1 and str2 relatively. Used only for stopping recursive processes.)
  + **private int[,] FindMEDValues(string str1, string str2)** 🡪 Creates a matrix that includes all MED values between substring of st1 and str2.
  + **public string[,] GenerateGridData(string source, string target)** 🡪 Implemented to prepare data for UI. Just adds header and indexes to matrix taken from “FindMEDValues” function. Returns a generic data matrix to display it.
  + **static int MinIndex(int x, int y, int z)** 🡪 is a utility function, takes 3 integer and return minimum value.
  + **static int Min(int x, int y, int z)** 🡪 is a utility function takes 3 integer number and return index of minimum number.(0 for x, 1 for y and 2 for z)
  + **public List<string> FindOperations(string[,] data)** 🡪 Implemented to detect operation names that is done by algorithms. Takes data matrix that includes all med data generated by “GenerateGridData” function and return an operations list.
* **public partial class** Form1 includes functions and procedures only for UI and test purposes and each of them explained blow;
  + **private void findAlternativesButton1\_Click(object sender, EventArgs e)** 🡪 Runs the processes of button named “Find Alternates”. Calls “FindAlternates” function from the MED class. Uses a stopwatch to diagnose running time.
  + **private void medValuesButton\_Click(object sender, EventArgs e)** 🡪 Runs the processes of “MED Values” button. Calls the “GenerateGridView” function and populates the grid view of UI. Uses a stop watch just like other button explained.
  + **private void PopulateGridView(Object[,] data)** 🡪 is a utility procedure that fill each cell of grid view with taken data matrix.
  + **private void WriteOperationList(string[,] data)** 🡪 Calls “FindOperations” function and uses the returned list and writes it to text box in UI.

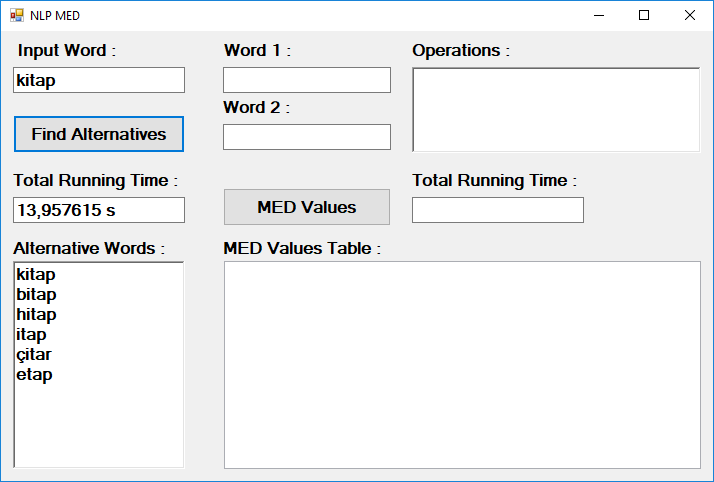
1. **Screenshots of Program**
   * **General View of the Program UI**

Left side is the Part 1 of the assignment, user enters word, clicks find alternatives button then program displays the 5 alternative word and total running time.

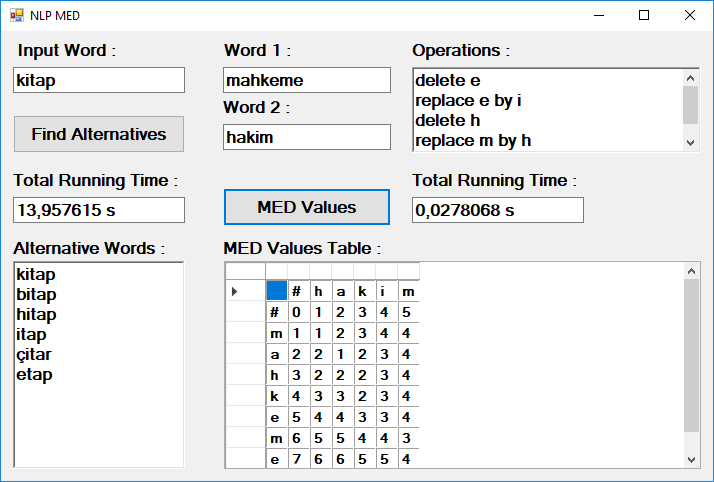
Right is the Part 2 of the assignment, user enters two word and program displays the med value table, operation names and total running time.



* **Part 1 Finding Alternative Words**



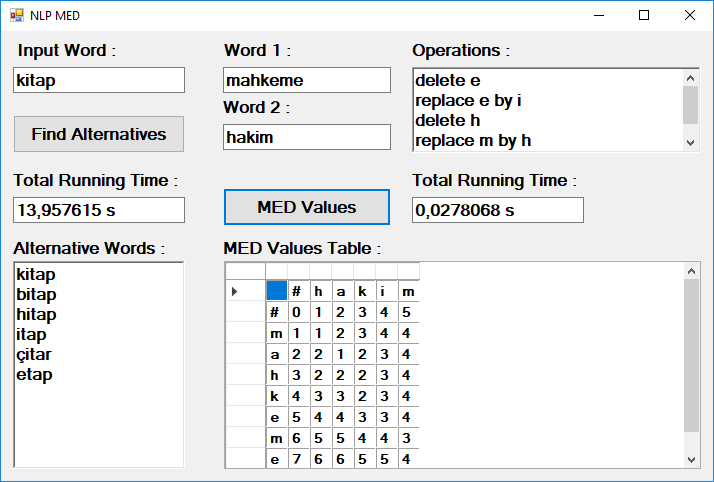
* **Part 2 Finding MED Values**



1. **Test Results**

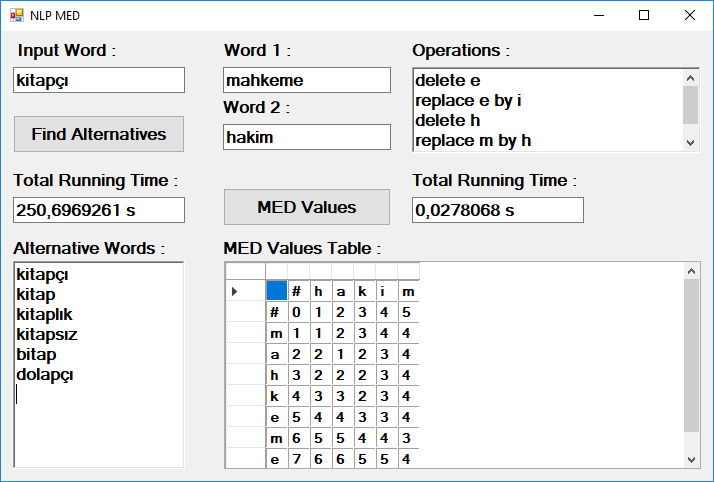
* **Part 1 : Find Alternatives**

Total Running Time for word “kitap” is 13,957615 second.



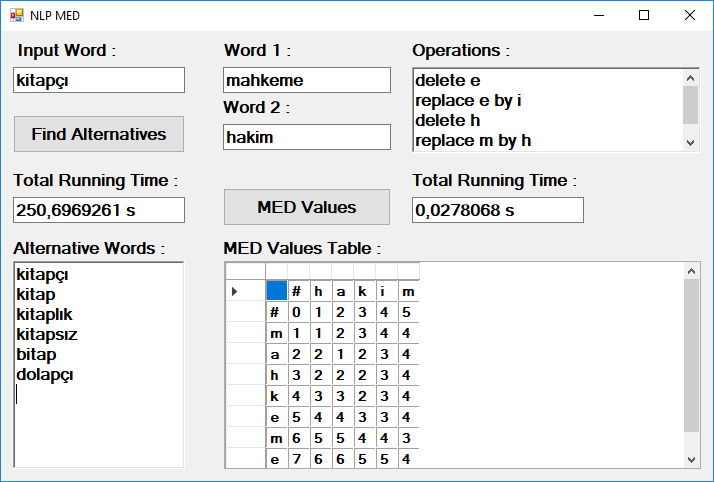
Running time depends on length of the input word. Total running time exponentially increases by the input length.

For example, running time for the word “kitapçı” is 250,6969261 second.

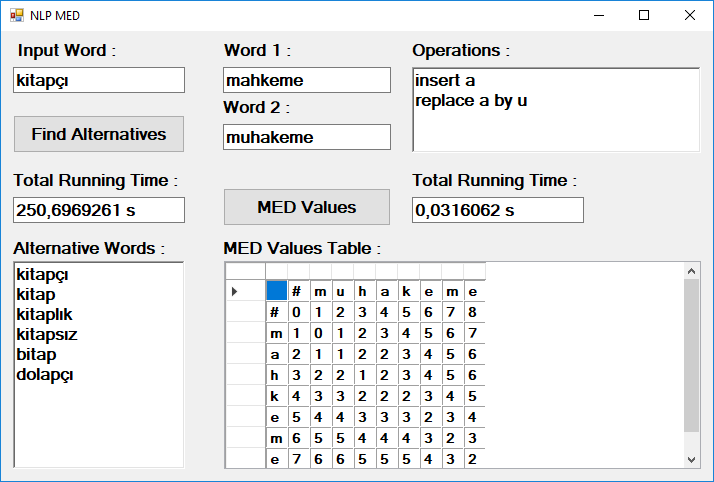


* **Part 2 : MED Values**

Total Running Time for words “mahkeme” and “hakim” is 0,0278068 second.



Total running time for words “mahkeme” and “muhakeme” is 0,0316062 second.



1. **Version-2 Notes**

Previously, program executable file and dictionary file have to be in the same folder to run the program without any error. In version-2 user can define dictionary file path independent with executable file. (But by default, program looks same folder, if can not finds then waits for user input). User unable to interact button until dictionary file successfully initialized. Second addition: Table now shows the operations path with bold font numbers.

Added functions;

**private void InitFiles()** 🡪 Initializes MED object with a path string.

**private void changeDictionarybutton1\_Click(object sender, EventArgs e)** 🡪 Change dictionary button handler. Just calls InitFiles function.

