CS300 – Spring 2018-2019 - Sabancı University

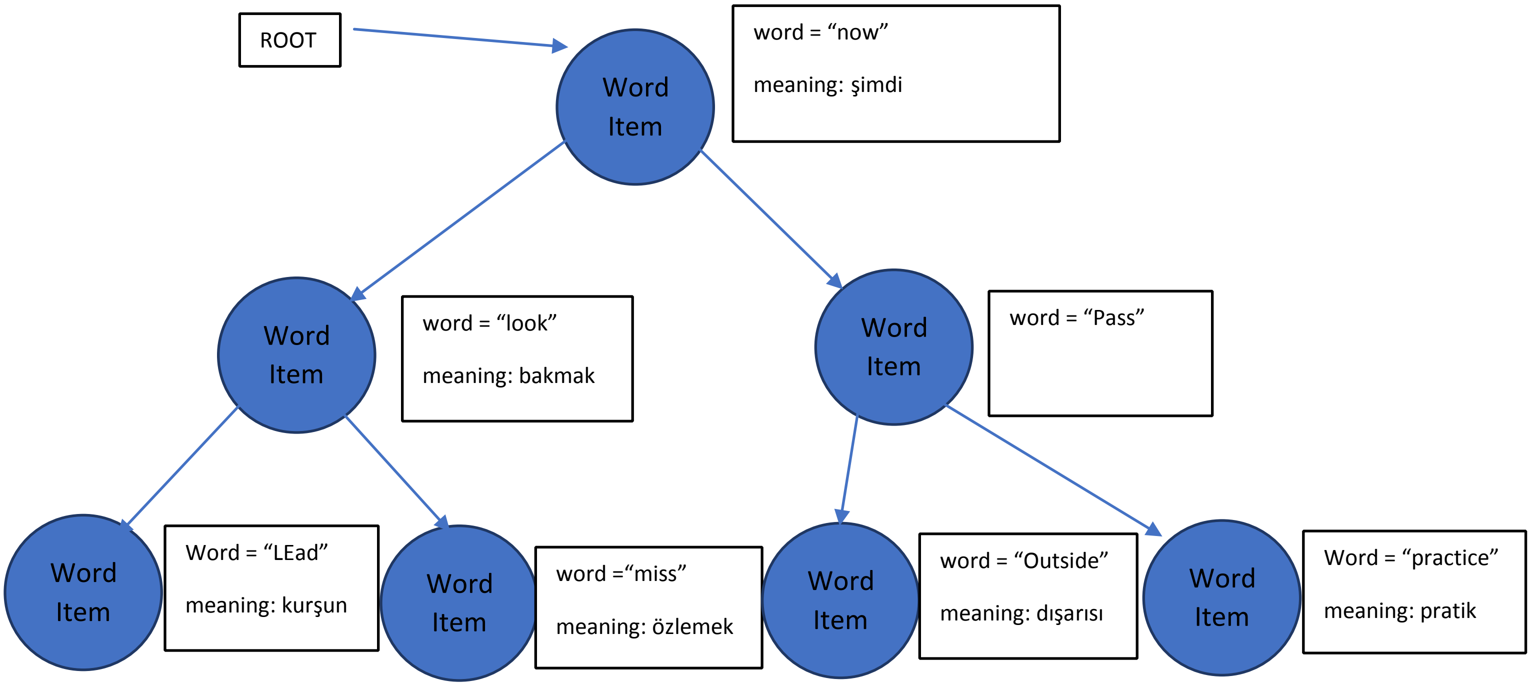
Homework #2 – Dictionary

Due March 23 Saturday at 22:00

Brief Description

In this homework, you are going to implement an English-to-Turkish dictionary. The dictionary will have hundreds of words to find the translations of a given word from the user. Also, the dictionary will let you add new translations or update the current translation. Obviously, in order to make translation fast, it is necessary to use a data structure with a fast search algorithm such as a Binary Search Tree (BST). Thus, you are required to use a BST for the implementation of your dictionary in this homework.

We provide you with a collection of words and their translations in the *dict.txt* file. You need to preprocess this file and construct your BST. For each word from the file, you need to insert a node into your BST. Of course you also need to keep track of the word's translations. So, you need to implement a templated BST first.



Program Flow

Program start with reading “dict.txt” file. Each line in the file provides an English word with its translation. Note that there might be empty lines, so you need to somehow skip them. The words in each line separated by spaces. There is a possibility of having more than one space between each word, and also there may be some words without any translations, and also the program is not case sensitive.

Note that, this “dict.txt” file is given to you **ONLY** for debugging purpose. We will use different files to grade your submissions.

Input / Output format

The input will be given from the user. There are three different commands in our dictionary. The commands are listed below:

- 1 : Query specific word: gets a word from user and shows it’s translation;
- 2: Add new translation for existing word: checks whether the word exists or not, if it exists program gets translation from user, otherwise shows message that word does not exist;
- 3: Add new word: checks whether the word exists or not, if it exists the program shows message that word exists, otherwise program gets word and its translation from user;

Some example runs may be given as follows (note that the red ones are the inputs):

Sample Run 1:

```
***Help***
Query: 1
Add new translation: 2
Add new word: 3
Exit: 0
Enter Command: 1
Enter queried word: trip
trip ---> yolculuk
Enter Command: 5
Enter Command: we'll
Enter Command: 1
Enter queried word: we'll
we'll ---> biz (gelecek zaman)
Enter Command: 0
Bye...
```

Sample Run 2:

```
***Help***
Query: 1
Add new translation: 2
Add new word: 3
Exit: 0
Enter Command: 2
Which word do you want to edit: love
Enter the new translation: aşk
***UPDATED***
Enter Command: 1
Enter queried word: love
love ---> sevmek, aşk
Enter Command: 1
Enter queried word: space
space ---> uzay, boşluk
Enter Command: 2
Which word do you want to edit: space
Enter the new translation: mekan
***UPDATED***
Enter Command: 1
Enter queried word: space
space ---> uzay, boşluk, mekan
Enter Command: 0
Bye...
```

Sample Run 3:

```
***Help***
Query: 1
Add new translation: 2
Add new word: 3
Exit: 0
Enter Command: 3
Enter New Word: love
Word already exists, if you want to add new translation please use command 2.
Enter Command: 3
Enter New Word: contemporary
Enter Translations: çağdaş
Enter Command: 1
Enter queried word: contemporary
contemporary ---> çağdaş
Enter Command: 0
Bye...
```

General Rules and Guidelines about Homeworks

The following rules and guidelines will be applicable to all homeworks, unless otherwise noted.

How to get help?

You may ask questions to TAs (Teaching Assistants) of CS300. Office hours of TAs are at the syllabus. Recitations will partially be dedicated to clarify the issues related to homework, so it is to your benefit to attend recitations.

What and Where to Submit

Please see the detailed instructions below/in the next page. The submission steps will get natural/easy for later homeworks.

Grading and Objections

Careful about the semi-automatic grading: Your programs will be graded using a semi-automated system. Therefore, you should follow the guidelines about input and output order; moreover, you should also use same prompts as given in the Sample Runs. Otherwise semi-automated grading process will fail for your homework, and you may get a zero, or in the best scenario you will lose points.

Grading:

- Late penalty is 10% off the full grade and only one late day is allowed.
- **Having a correct program is necessary, but not sufficient to get the full grade. Comments, indentation, meaningful and understandable identifier names, informative introduction and prompts, and especially proper use of required functions, unnecessarily long program (which is bad) and unnecessary code duplications will also affect your grade.**
- Please submit your own work only (even if it is not working). It is really easy to find out “similar” programs!
- For detailed rules and course policy on plagiarism, please check out http://myweb.sabanciuniv.edu/gulsend/courses/cs201/plagiarism/

Plagiarism will not be tolerated!

Grade announcements: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

Grade objections: It is your right to object to your grade if you think there is a problem, but before making an objection please try the steps below and if you still think there is a problem, contact the TA that graded your homework from the email address provided in the comment section of your announced homework grade or attend the specified objection hour in your grade announcement.

- Check the comment section in the homework tab to see the problem with your homework.
- Download the .zip file you submitted to SUCourse and try to compile it.
- Check the test cases in the announcement and try them with your code.

- Compare your results with the given results in the announcement.

What and where to submit (IMPORTANT)

Submissions guidelines are below. Most parts of the grading process are automatic. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.

Add your name to the program: It is a good practice to write your name and last name somewhere in the beginning program (as a comment line of course).

Name your submission file:

- Use only English alphabet letters, digits or underscore in the file names. Do not use blank, Turkish characters or any other special symbols or characters.
- Name your cpp file that contains your program as follows.
  “**SUCourseUserName_yourLastname_yourName_HWnumber.cpp**”
- Your SUCourse user name is actually your SUNet user name which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsızkodyazaroğlu, then the file name must be:
  cago_ozbugsizkodyazaroglu_caglayan_hw2.cpp
- Do not add any other character or phrase to the file name.
- Make sure that this file is the latest version of your homework program.
- You need to submit ALL .cpp and .h files including the robot and minifw files in addition to your main.cpp in your VS solution. Compress all your necessary files using WINZIP or WINRAR programs. Please use "**zip**" compression. "rar" or another compression mechanism is NOT allowed. Our homework processing system works only with zip files. Therefore, make sure that the resulting compressed file has a zip extension.
- Check that your compressed file opens up correctly and it contains your **cpp** file. You will receive no credits if your zip file does not expand or it does not contain the correct file.
- The naming convention of the zip file is the same as the cpp file (except the extension of the file of course). The name of the zip file should be as follows.
  “**SUCourseUserName_yourLastname_yourName_HWnumber.zip**”
  For example zubzipler_Zipleroglu_Zubeyir_hw3.zip is a valid name, but hw2_hoz_HasanOz.zip, HasanOzHoz.zip are NOT valid names.

Submission:

- Submit via SUCourse ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).
  1. Click on "Assignments" at CS300 SUCourse.
  2. Click Homework 1 in the assignments list.
  3. Click on "Add Attachments" button.
  4. Click on "Browse" button and select the zip file that you generated.
  5. Now, you have to see your zip file in the "Items to attach" list.
  6. Click on "Continue" button.
  7. Click on "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

Resubmission:

- After submission, you will be able to take your homework back and resubmit. In order to resubmit, follow the following steps.

1. Click on "Assignments" at CS300 SUCourse.
2. Click Homework 3 in the assignments list.
3. Click on "Re-submit" button.
4. Click on "Add/remove Attachments" button
5. Remove the existing zip file by clicking on "remove" link. This step is very important. If you don't delete the old zip file, we get both files and the old one may be graded.
6. Click on "Browse" button and select the new zip file that you want to resubmit.
7. Now, you have to see your new zip file in the "Items to attach" list.
8. Click on "Continue" button.
9. Click on "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!
Omid Khorsand Kazemy and Gülşen Demiröz