(S2-21_DSECLZG519)
(Data Structures and Algorithms Design)
Academic Year 2021-2022

Assignment 1 – PS04 - [Palindrome] - [Weightage 12%]

1. Problem Statement

A phrase is a palindrome, that is, it reads the same forward and backward (for example, the word "rotator'" is a palindrome, as is the string "rats live on no evil star"). Write a program that employs two stacks in order to determine if a phrase is a palindrome. Read the characters one by one and transfer them into a stack. The characters in the stack will then represent the reversed word. Once all characters have been read, transfer half the characters from the first stack into a second stack. Thus, the order of the words will have been restored. If there were an odd number of characters, remove one further character from the original stack. Finally, test the two stacks for equality, element by element. If they are the same, then the word/phrase is a palindrome.

Requirements

1. Implement the above problem statement using Python 3.7.
2. Read the input from a file (inputPS04.txt) and separate each word from it and input it to a stack.
3. You will output your answers to a file (outputPS04.txt) where the file contains only the result (Palindrome: yes, Palindrome :No).
4. Perform an analysis for the features above and give the running time in terms of input size: n. (n words are in the input file inputPS04.txt)

Sample Input:

 A nut for a jar of tuna

Input will be taken from the file(inputPS04.txt) here each word is a space-separated as shown in the example above

Sample Output:

 A nut for a jar of tuna

If the word/phrase is a palindrome, the output file should contain the actual word / phrase. If not, the output file should contain the message "Not a Palindrome"

Note that the input/output data shown here is only for understanding and testing, the actual file used for evaluation will be different

2. Deliverables

1. Word document **designPS04_<group id>.docx** detailing your design and time complexity of the algorithm and a contribution table as below

2. **inputPS04.txt** file used for testing

3. **outputPS04.txt** file generated while testing

4. .py file containing the python code. Create a single python file. Do not fragment your code into multiple files

5. Zip all of the above files including the design document in a folder with the name: **[Group id] _A1_PS04_Palindrome.zip** and submit the zipped file.

   Group Id should be given as Gxxx where xxx is your group number. For example, if your group is 26, then you will enter G026 as your group id.

3. Instructions

1. It is compulsory to make use of the data structure(s) / algorithms mentioned in the problem statement.

2. Ensure that all data structure insert and delete operations throw appropriate messages when their capacity is empty or full. Also ensure basic error handling is implemented.

3. For the purposes of testing, you may implement some functions to print the data structures or other test data. But all such functions must be commented before submission.

4. Make sure that your read, understand, and follow all the instructions

5. Ensure that the input, prompt and output file guidelines are adhered to. Deviations from the mentioned formats will not be entertained.

6. The input, prompt and output samples shown here are only a representation of the syntax to be used. Actual files used to evaluate the submissions will be different. Hence, do not hard code any values into the code.

7. Run time analysis is to be provided in asymptotic notations and not timestamp based runtimes in sec or milliseconds.

8. Please note that the design document must include

9. The data structure model you chose with justifications

10. Details of each operations with the time complexity and reasons why the chosen operations are efficient for the given representation

11. One alternate way of modelling the problem with the cost implications.

12. Writing good technical report and well document code is an art. Your report cannot exceed 4 pages. Your code must be modular and quite well documented.

Instructions for use of Python:

1. Implement the above problem statement using Python 3.7.
2. Use only native data types like lists and tuples in Python, do not use dictionaries provided in Python. Use of external libraries like graph, numpy, pandas library etc. is not allowed. The purpose of the assignment is for you to learn how these data structures are constructed and how they work internally.
3. Create a single *.py file for code. Do not fragment your code into multiple files.
4. Do not submit a Jupyter Notebook (no *.ipynb). These submissions will not be evaluated.
5. Read the input file and create the output file in the root folder itself along with your .py file. Do not create separate folders for input and output files.

4. Deadline

1. The strict deadline for submission of the assignment is Tuesday, **21st June, 2022**.
2. The deadline has been set considering extra days from the regular duration in order to accommodate any challenges you might face. No further extensions will be entertained.
3. Late submissions will not be evaluated.

5. How to submit

1. This is a group assignment.

2. Each group has to make one submission (only one, no resubmission) of solutions.

3. Each group should zip all the deliverables in one zip file and name the zipped file as mentioned above.

4. Assignments should be submitted via Canvas > Assignment section. Assignment submitted via other means like email etc. will not be graded.

6. Evaluation

1. The assignment carries 12 Marks.
2. Grading will depend on
   a. Fully executable code with all functionality working as expected
   a. Well-structured and commented code
   b. Accuracy of the run time analysis and design document.
3. Every bug in the functionality will have negative marking.

4. Marks will be deducted if your program fails to read the input file used for evaluation due to change / deviation from the required syntax.

5. Use of only native data types and avoiding libraries like numpy, graph and pandas will get additional marks.

6. **Plagiarism will not be tolerated. Copy / Paste's from web resources / or your friends' submission will attract severe penalty to the extent of awarding 0 marks. We will not measure the extent of such blatant copy pastes and details of who copied from whom and such details while awarding the penalties. It's the responsibility of the team to solve and protect your original work.**

7. Source code files which contain compilation errors will get at most 25% of the value of that question.

7. Readings

Text book: Algorithms Design: Foundations, Analysis and Internet Examples Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition). **Chapters:** 2.1