

Task Sheet 2: Stack Based Problems

Level – Credit (This task would require the students to explore the concepts of Stacks in detail)

SIT221 Data Structures and Algorithms - Practical Task

Task Objective

In this task, you will create algorithms and data structures for stacks to handle certain challenges. You will provide a written report that includes a full description of your solutions and how they fit the requirements.

Background

Stacks are a basic data structure in computer science, distinguished by its Last-In-First-Out (LIFO) behaviour, in which the last piece added is the first to be deleted. Stacks offer two fundamental operations: Push (adding an element) and Pop (removing the top element). This work delves into sophisticated stack operations and applications, such as ensuring balanced parentheses and evaluating expressions using Reverse Polish Notation (RPN).

This assignment also includes improving a basic stack to handle FindMiddle and DeleteMiddle operations while keeping consistent $O(1)$ time complexity for all operations. Additional data structures are required to efficiently track the middle element. Understanding these sophisticated stack operations and their efficient implementation is critical for creating strong algorithms capable of handling complicated data manipulations while guaranteeing syntactical correctness and effective expression evaluation.

Task Requirements

Part 1: Pseudocode Implementation

1. Create a Stack With a Middle Operation:

Problem: A stack data structure supports Push and Pop, two operations for writing and reading data, respectively. Using the stack as a starting point, create a data structure that includes the FindMiddle and DeleteMiddle procedures, which return and delete the middle member of the stack, respectively. The new data structure must perform all four operations in $O(1)$ time.

2. Design an Expression Evaluator using a Stack:

Problem: Create a method to assess a mathematical equation written in Reverse Polish Notation. The expression will be a string of tokens separated by spaces. The method should support the four fundamental arithmetic operations: addition, subtraction, multiplication, and division.

Requirement: The algorithm's time complexity is $O(n)$, where n is the number of tokens in the statement.

Part 2: General Questions

Question 1:

Compare and contrast how stacks are used to evaluate infix, postfix, and prefix expressions.

Question 2:

Explain how to utilise a stack to construct the depth-first search (DFS) technique in graph traversal.

Question 3:

Describe a scenario in computer systems where a stack overflow might occur and how it can be avoided.

Further Notes

- You may find exploring chapters 6.1 and 6.2 of the course book "Data Structures and Algorithms in Java" by Michael T. Goodrich, Roberto Tamassia, and Michael H. Goldwasser (2014) useful to solve these tasks. You may access the book on-line for free from the reading list application in CloudDeakin available in Resources ® Additional Course Resources ® Resources on Algorithms and Data Structures ® Course Book: Data structures and algorithms in Java.

Marking Process and Discussion

To get your task completed, you must finish the following steps strictly on time.

1. Work on your task either during your allocated lab time or during your own study time.
2. Once the task is complete you should make sure that your program implements all the required functionality, is compliant, and has no runtime errors. Programs causing compilation or runtime errors will not be accepted as a solution. You need to test your program thoroughly before submission. Think about potential errors where your program might fail. Note we can sometime use test cases that are different to those provided so verify you have checked it more thoroughly than just using the test program provided.
3. Submit your solution as an answer to the task via the OnTrack submission system. This first submission must be prior to the submission "S" deadline indicated in the unit guide and in OnTrack.
4. If your task has been assessed as requiring a "Redo" or "Resubmit" then you should prepare a new submission. You will have 1 (7 day) calendar week from the day you receive the assessment from the tutor. This usually will mean you should revise the lecture, the readings indicated, and read the unit discussion list for suggestions. After your submission has been corrected and providing it is still before the due deadline you can resubmit.
5. If your task has been assessed as correct, either after step 3 or 4, you can "discuss" with your tutor. This first discussion must occur prior to the discussion "D".
6. Meet with your to demonstrate/discuss your submission. Be on time with respect to the specified discussion deadline.

7. The tutor will ask you both theoretical and practical questions. Questions are likely to cover lecture notes, so attending (or watching) lectures should help you with this compulsory interview part. The tutor will tick off the task as complete, only if you provide a satisfactory answer to these questions.
8. If you cannot answer the questions satisfactorily your task will remain on discussion and you will need to study the topic during the week and have a second discussion the following week.
9. Please note, due to the number of students and time constraints tutors will only be expected to mark and/or discuss your task twice. After this it will be marked as a “Exceeded Feedback”.
10. Note that we will not check your solution after the submission deadline and will not discuss it after the discussion deadline. If you fail one of the deadlines, you fail the task and this reduces the chance to pass the unit. Unless extended for all students, the deadlines are strict to guarantee smooth and on-time work through the unit.
11. Final note, A “Fail” or “Exceeded Feedback” grade on a task does not mean you have failed the unit. It simply means that you have not demonstrated your understanding of that task through OnTrack.

All Answers for this Task sheet:

https://drive.google.com/drive/folders/1hMp6VwAOSRM4QfVmJwKQcmIUoJD_9ZJ5?usp=share_link