Curtin College-Bentley
DSA1002 Data Structures and Algorithms
Trimester 2, 2022

Assignment 2

Weight: 50% of the unit

Assignment Location: The assignment 2 is uploaded under **Assessments section (Assessment 3 – Final Assessment)** on unit Moodle page**.**

Answer Format. When you write an answer, clearly indicate the relevant question number/letter. Include your name and student ID at the start. Also add appropriate comments to code files to indicate author name and student ID. **Please refer to detailed submission guidelines in section 2**.

Timeframe. You have 4 days (96 hours) to complete and submit your answers, from **12:00pm on 19th Sep 2022 until 12:00pm on 23rd Sep 2022 (UTC+8)**. You may schedule your work within this period. **However, late submissions are not allowed**.

Submission Location. Submit your answer document(s) to the "**Assessment 3 – Final Assessment**" area on Moodle under assessments section. You must verify that your submission was successful. **Correctly submitting is entirely your responsibility.**

Reference Material. This is an OPEN BOOK and OPEN COMPUTER assignment. You may refer to any written material, including your notes, course materials, books, websites, Unit Moodle page recordings etc. However:

- You must complete this assignment entirely on your own. You should answer all questions in your own words and code.
- You can use pseudo code and algorithms provide in the unit slides (Moodle page) for your implementation.
- During the assignment, you may not communicate with any other student regarding the test.
- During the assignment, you may not communicate with any other person in order to seek or receive an answer to any test question.
- Your answer document will be checked by text matching software for signs of cheating,collusion and/or plagiarism.
- The assignment questions have been designed such that any two students, working independently,should not produce the same answers.
- The coding part of this assignment can be submitted in either python or java.
- **All parts taken from your own previous submissions (practical/assignment) should also need to be referenced as per college recommended citation style**.

1. Overall Assignment Description

In this assessment you will apply a detailed knowledge of data structures and algorithms to real-life applications to show your understanding of the details covered in the unit. Detailed description on each question and steps required to be performed can be found in the question description.

Question 1: General Understanding (Total Marks: 10)

A. Consider a situation where you need to store the information of friends list of users on a social networking application. The list of friends is implemented as a dynamic data structure, where friends can be added or removed when required. The data structure should be able to track the mutual friends of multiple users, where connections between users are random and do not have any specific priority/order.

   i. Describe your selection of the ADT to store individual user record. **(3 marks)**

   ii. Discuss the effectiveness of your selection for maintaining the friends list and mutual friends list determination. **(3 marks)**

B. The student record management software needs to store the information of thousands of students at a time. The information of each student is sorted based on the student ID. Each student is tracked by his/her ID. This system also requires quick student record searching regardless of number of students (data size) for adding, removing, editing and updating the record.

   i. Support selection of a particular ADT based on the asymptotic time complexities (Big O notation) for record insertion, removal, traversal and record printing. **(4 marks)**

Question 2 on Next Page

Question 2: Recursion and Sorting (Total Marks: 16)

A. Consider the situation where you are supposed to sort data of all students (on student ID) in Curtin College/University (approximately 50,000 students or more). Support your selection of two best sorting algorithms describing stable vs unstable and in-place vs non-in-place property of the selected sorts. **(4 marks)**

B. Consider the application of merge sort for low memory tablet-based computer for sorting. Will this be good choice of the described application, justify your answer by considering O(NlogN) time complexity and memory required to execute the sort. **(4 marks)**

C. Presence of redundant data elements (i.e., duplicate numbers in a list) could affect the execution of a sorting algorithm, if yes how it can affect the complexity of sorting. **(3 marks)**

D. Convert the following iterative pseudo code (Fig. 1) to the recursive python code. Your code must include wrapper method and exception handling. **(5 marks) [clearly mention the base case in comments and provide basic test harness (simple main function in the same file)]**

```
Function revrse_str
Imports: Str1
Exports: Reverse Str1
for x =size (Str1)-1 to 0
      print (Str1(i))
End for
End
```

Fig. 1

Question 3 on Next Page

Question 3: Stacks, Queue and Lists (Total Marks: 15)

A. Implement a function "Transfer (S1,S2, S3)" where S1, S2 and S3are three stacks. The function transfers the data from S1 to S2 and S2 to S3, at the end print S1, S2 and S3 are printed to show change in the order of the elements **(5 marks) [You can use numpy arrays or linked list for this implementation]**

B. Implement a circular queue (max size 10 elements) with the help of double ended single linked list, try to insert 12 random names of fruits in the queue, however the queue should be able to throw appropriate error messages/exceptions if the queue is full or empty. **(5 marks)**

C. Discuss in detail the selection of double ended double linked list compared to double ended single linked list, single ended single linked list and array for the implementation of circular queue (Part B). Discuss in detail that which data structure can be used for this implementation, effective on time complexity and memory utilisation Consider the Big O notation (asymptotic) time complexity and memory utilisation to justify your answer. **(5 marks) [can be represented in tabular form]**

Question 4 on Next Page

Question 4: Trees (Total Marks: 16)

A. Considered the list/array shown in Fig. 2 and insert the elements in a Red Black Tree, show the developed tree. Also, draw the Red Black tree and BST on the reverse sorted data. Now compare the developed tree (all three) describe what is the effect of sorted or partially sorted data on the trees. Which is the best tree if compared on completeness. **(4 marks) [show steps for each element insertion]**

43	61	-2	4	15	91	32	21	11	51	22	14	35	8	87	78	29

Fig. 2

B. Consider the elements presented in Fig. 3 and draw a 2-3-4 Tree form initially empty root. Show and describe each step. **(3 marks)**

10	5	12	7	101	25	17	16	32	33	11	3	71	26	1	22	56	86	15	-9

Fig. 3

C. Convert the tree drawn in part B to a B Tree (degree 5), also discuss that which tree is more efficient for inserting and searching elements. Consider the Big O notation (asymptotic) time complexity to support you answer. **(4 marks)**

D. Consider the following UML diagrams (Fig. 4) for the BSTree class and BSTNode class. You are supposed to write a method to print all entries divisible by 4 and 5 in a sorted order without using any explicit sorting algorithm (i.e., any sorting algorithm). Write a test harness to check the functionality of the code. **(5 marks)**

<<BSTNode>>
-Key: None -Value: None -Left_child: None -Right_child: None
+init(self, key, value)

<<BSTree>>
-Root: None
+init(self, root) +sorted_4_5_num(self)

Fig. 4

Question 5 on Next Page

Question 5: Graphs (Total Marks: 12)

***Consider the Fig. 5 for the following questions.**

A. Represent the following graph in Adjacency Matrix and Adjacency List (for each vertex) representation. **(4 marks)**

B. If this graph is used to represent World Wide Web (WWW), considering this application, describe that what data should be stored in vertices (smallest entity is each webpage) and edges respectively. **(2 marks)**

C. Described the graphical steps for the depth first search along with value stored in stack/queue. Consider vertex "G" as the starting point. **(3 marks)**

D. Described the graphical steps for the breadth first search along with value stored in stack/queue. Consider vertex "D" as the starting point. **(3 marks)**

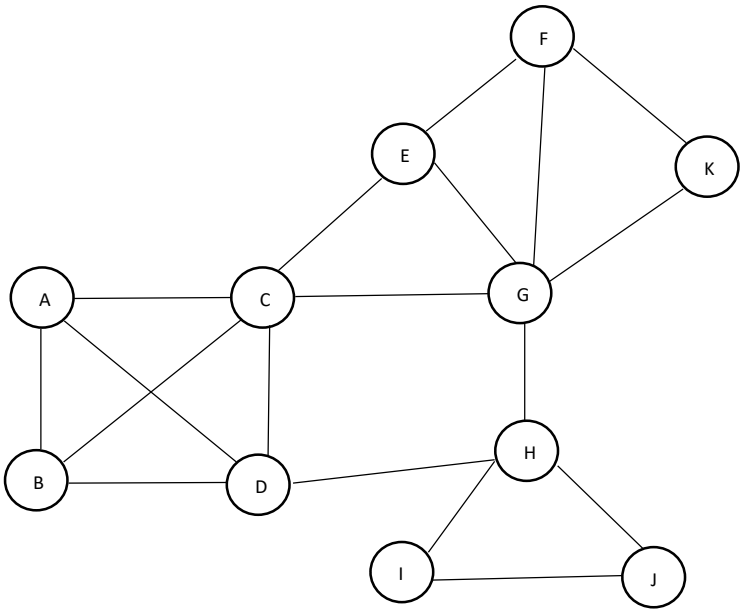

Fig.5

Question 6 on Next Page

Question 6: Heaps (Total Marls: 13)

*Consider the number list presented in Fig. 6 for this question.

A. Draw a max heap using the number list, show you working as graphical steps for each insert, representing the step by step building of the heap. **(4 marks)**

B. Show an array-based representation of the number list as heap, satisfying the related arithmetic operations for traversing. **(2 marks)**

C. Show the process for deleting/removing 2 values in the tree built in part A. Show steps (including each step of the trickle-down) on the array-based representation built in part B. **(4 marks)**

D. Give an example of real-life applications of Heap data structure, clearly describe how the data will be stored (i.e., identify key). **(3 marks)**

25	11	19	5	-4	0	58	87	98	109	55	41	69	32	15	20	7	2	8

Fig. 6

Question 7 on Next Page

Question 7: Hash Table(s) (Total Marks: 18)

A. Consider the following (Fig.7) hash functions for implementation of quadratic probing with a table length of 27 items. Use the hash function to insert the following value (Fig.8) in the hash table (add missing rows). Show your steps including any collusions. **(6 marks)**

B. How you will keep the load factor below 0.75 for the hash table, what exactly is the meaning of using this load factor in the case of the hash table developed in Part A. Describe your answer by stating effectiveness w.r.t. memory utilisation and overhead of hash table resizing for this load factor selection. **(6 marks)**

C. Describe the limitations of separate chaining for the implementation of a hash table, describe how the searching and element insertion time is affected by the separate chain-based implementation of hash table. Show and explain the time complexity for separate chaining. **(6 marks)**

Hash function 1

```
def hash1(key):
    checksum = 3445
    for i=1 to length(key):
        ch = key[i]
        checksum = checksum +ASCII(ch)
    hashIdx = checksum%27
    return hashIdx
```

Fig. 7

Data to Insert

Key	Value
"24"	"Student 1"
"31"	"Student 2"
"1234"	"Student 3"
"465"	"Student 4"
"789"	"Student 5"
"890"	"Student 6"
"93"	"Student 7"

Index	Hash Table	
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
..		
..		
26		

Fig. 8

Submission Guidelines on Next Page

2. Submission Guidelines

You will be provided **two separate submission points for code**(s) and report, make sure to submit the code and report in the respective submission points. Consider the following while submitting:

a. The report should be submitted as a word/pdf document, clearly mentioning the question number for the respective answer.

b. Code(s) should be submitted as a single compressed zip/tar.gz file (named as DSA1002_A2_StudID), name the code files (in the folder) appropriately e.g., "Q1PartA.py"

Please verify that your submission is correct and not corrupted. You may make multiple submissions, only your last one will be marked. **However, late submissions are strictly not allowed.**

3. Academic Integrity

Please see the Coding and Academic Integrity Guidelines on unit Moodle page.

In summary, this is an assessable task. If you use someone else's work or assistance to help complete part of the assignment, where it's intended that you complete it yourself, you will have compromised the assessment. You will not receive marks for any parts of your submission that are not your own original work. Further, if you do not reference any external sources that you use, you are committing plagiarism and/or collusion, and penalties for academic misconduct may apply.

Curtin college also provides general advice on academic integrity at https://www.curtincollege.edu.au/content/dam/navitas/upa/curtin/pdfs/academic-integrity-policy.pdf

The unit coordinator may require you to provide an oral justification of, or to answer questions about, any piece of written work submitted in this unit. Your response(s) may be referred to as evidence in an academic misconduct inquiry.