Curtin College-Singapore
DSA1002 Data Structures and Algorithms
Trimester 3, 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 **10:00am on 24th Jan 2023 until 10:00am on 28th Jan 2023 (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. A car manufacturing company need to store data (permanently i.e., hard drive) of each individual car including model, year of manufacturing, colour, variant and engine size. You are supposed to compare the data structures (listed below) and suggest one which is best suited for the said application. The complexity of insertion (single record), sorting and searching should be compared to make the selection. Also, as the data is stored on an external storage (hard drive) please consider which data structure will be best suited for reading data in terms of blocks (hard drive data blocks).

   i. Describe the way you will store the data of an individual car i.e., ADT (see the details of a car described above). (**2 marks**)

   ii. Compare the following data structures and select the best suited for the application. [Hint: you can make a table for comparison] (**4 marks**)

      a. Hash Table

      b. 2-3-4 tree

      c. B Tree (Block Tree)

B. Curtin college wants to store the weekly schedule for a year 2023, the schedule should include room number, seating capacity, count of workstations and classes scheduled in the room. You should suggest the data structure best suited to store the information such that following operations are supported (can be performed effectively/efficiently) on the schedule (application).

   a. Adding new rooms (records) based on room number as key

   b. Deleting old records

   c. Searching a room (room number as key)

   i. Which data structure you will use if want to make the adding, deleting time complexity independent of the data size. Support your selection with appropriate comments on time complexities. (**2 marks**)

   ii. What is memory overhead (i.e., any extra memory used than the data size) of your selected data structure.

* If you are defining your ADT in terms of classes best to represent is UML class diagram(s).

Question 2 on Next Page

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

A. You have been provided with 1 million random number (integer type). You are supposed to suggest (among the listed below) the best sorting algorithm based on the data provided. Make clear comments considering stable/non-stable and in-place/not in-place properties and time complexity of each sort. (**4 marks**)

   a. Selection sort

   b. Quick sort

   c. Counting sort

   d. Radix Sort

B. Consider the application of quick sort for low computation power tablet-based computer for sorting. Will this be good choice of the described application, justify your answer by considering time complexity, in-place/not in-place property of the sorting algorithm. **(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 DSAFun
Imports: root: Root of BST
Exports: None
Curr := root
while (True) do
  if Curr :≠ None
     Stack.push(Curr)
    Curr = Curr → left
 if stack.pointer :≠
   value = Stack.pop()
   print(value)
  Curr = Curr → right
 else
   break
end while
```

Fig. 1

Question 3 on Next Page

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

A. Implement a function "Transfer (Q1, Q2, Q3)" where Q1, Q2 and Q3 are three circular queues. The function transfers the data from Q1 to Q2 and Q2 to Q3, at the end print Q1, Q2 and Q3 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 stack (max size 10 elements) with the help of double ended single linked list, try to insert 12 random names of cars in the stack, however the stack should be able to throw appropriate error messages/exceptions if the stack 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 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)

* All values given in the figures below should be used/read left to right.

A. Considered the list/array shown in Fig. 2 and insert the elements in a Binary Search Tree, show the developed tree. Also, draw the Red Black tree on the reverse sorted data. Now compare the developed trees describe what is the effect of sorted or partially sorted data on the trees. Which is the best tree if compared on completeness and balance, support your answer with O notation complexity details **(4 marks)** **[show steps for each element insertion]**

52	86	14	-9	12	89	7	44	35	32	78	19	15	78	69	55	51	41	5	9	1	33	44	13	96	24	98

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)**

52	86	14	-9	12	89	7	44	35	32	78	19	15	78	69	55	51	41	5	9	1	33	44	13	96	24	98

Fig. 3

C. Convert the tree drawn in part B to a B Tree (degree 7), 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 elements (Fig 4) and provide an algorithm (Pseudo/python/java code) to read one value at a time and convert it into an array-based representation of a binary tree (Hint: the rules for parent, left and right child defined in heap can be used here). The array based binary tree representation should then be shown as a tree as well. (**5 marks**)

52	86	14	-9	12	89	7	44	35	32	78	19	15	78	69	55	51	41	5	9	1	33	44	13	96	24	98

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 connectivity of people on a social network, considering this application, describe that what data should be stored in vertices (smallest entity personal unique ID e.g., twitter handle) and edges respectively. **(2 marks)**

C. Described the graphical steps for the depth first search along with value stored in stack/queue. Consider vertex "5" 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 "11" as the starting point. **(3 marks)**

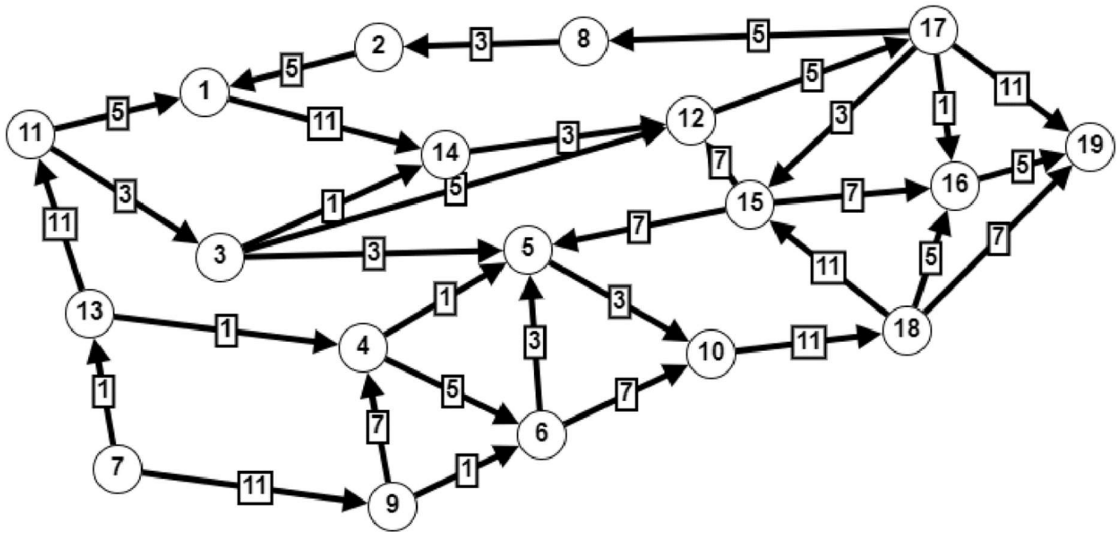

Fig.5

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 min heap (tree-based) using the number list, show you working as graphical steps for each insert, representing the step by step building of the heap. **(4 marks)** [Hint: you can use text to represent the steps]

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 two values one after the other (78 and -9) 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)** [Hint: you can use text to represent the steps]

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)**

22	16	17	-5	1	78	71	43	36	52	88	20	17	25	67	54	55	4	5	9	1	33	45	16	76	24	98

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 linear probing with a table length of 17 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 0.85 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. Consider each index in the table as memory location if the load factor is 0.75 many memory locations (approx) will remain empty for table size of 50K locations. **(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 = 11169
    for i=1 to length(key):
        ch = key[i]
        checksum = checksum +ASCII(ch) + 8
    hashIdx = checksum% table_size
    return hashIdx
```

Fig. 7

Data to Insert

Key	Value
"4"	"Student 1"
"2331"	"Student 2"
"34"	"Student 3"
"489"	"Student 4"
"799"	"Student 5"
"250"	"Student 6"
"930"	"Student 7"

Index	Hash Table	
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
..		
..		
16		

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.