Curtin College-Bentley
DSA1002 Data Structures and Algorithms
Trimester 1, 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 **08:00am on 27th May 2022 until 08:00am on 31st May 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/java.

1. Overall Assignment Description

In this assessment you will apply a detailed knowledge of data structure and algorithms to real-life application to have a better 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 parcels sent by a courier company. The parcels need to be organised in a date-based ordered record. The data of the company is large and need to be stored permanently. Considering the business model of the company the records need to be accessed frequently. The related operations on the parcels record also include searching and sorted data retrieval.

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

   ii. Discuss the effectiveness of your section for new data entry, searching and sorting. **(3 marks)**

B. The library book management software needs to store the information of thousands of books at a time. The books information is sorted based the books ISBN (International Standard Book Number). Any book issued to a student is tracked by its ISBN number for student name, ID and due date for returning. This system also requires quick books record traversing for books issuing, returning and management.

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

Question 2 on Next Page

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

A. Consider the situation where the students (consider max student count 30) of each unit at Curtin College need to be sorted based on their weekly attendance percentage for each week. You as an expert in algorithms are provided with an option of selection between quick sort and insertion sort. Justify your selection for the time complexity and memory utalisation. **(4 marks)**

B. You are working on an application which works on low memory and computational power tablets. Argument on utilisation of quick, merge, insertion and bubble sort for your application design. **(4 marks)**

C. Why selection of pivot is important and for quicksort, comment on the minimum and maximum value selected as pivot in a list. **(3 marks)**

D. Convert the following iterative pseudo code (Fig. 1) to the recursive python/java code. Your code must include wrapper method and exception handling. **(5 marks)**

```
Function vow_vs_con
Imports string
Exports vowels versus consonant: Boolean
for x =1 to n
        if string (x) in vowels
                v_count ← v_count + 1
        else
                c_count ← c_count +1
if v_count lessthan c_count
        result ← 1
return result
```

Fig. 1

Question 3 on Next Page

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

A. Implement a Stack (S) with a Top pointer (T) which push 10 random names of fruits, and pop them all with the help of non-recursive function (Pop()) while storing them in a list in the same order (from index 0) as they were entered in a stack initially. **(5 marks)**

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

C. Discuss in detail the selection of double linked list compared to simple list/array for the implementation of circular queue, would it be good or worse. Consider the Big O notation (asymptotic) time complexity and memory utalisation to justify you answer. **(5 marks)**

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 Binary Search Tree (BST) in the provided and sorted order (i.e. sort this array before inserting to BST). Compare both trees to describe the limitation of the BST data structure. Also describe which tree is efficient for traversal. **(4 marks)**

23	89	34	15	7	91	44	22	1	11

Fig. 2

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

3	1	5	4	2	9	10	8	7	6

Fig. 3

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

D. Consider the following python code (Fig. 4) code for the BST class definition. You are supposed to write a method to print every odd number entered in the tree 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)**

```
class BinarySearchTree():
    class BSTNode():
        def init(self, key, value):
            self.key = key
            self. value =  value
            self.left = None
            self.right = None
    def __init__(self):
        self.root = None
    def sorted_odd_nums(self):
            #your implementation
```

Fig. 4

Question 5 on Next Page

Question 5: Graphs (Total Marks: 10)

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

A. Represent the following graph in Adjacency Matrix and Adjacency List representation. **(2 marks)**

B. If this graph is used to represent multiple traffic routes in the city, considering this application, describe that what data should be stored in vertices and edges respectively. **(2 marks)**

C. Described the graphical steps for the depth and breadth first search along with value stored in stack/queue. **(6 marks)**

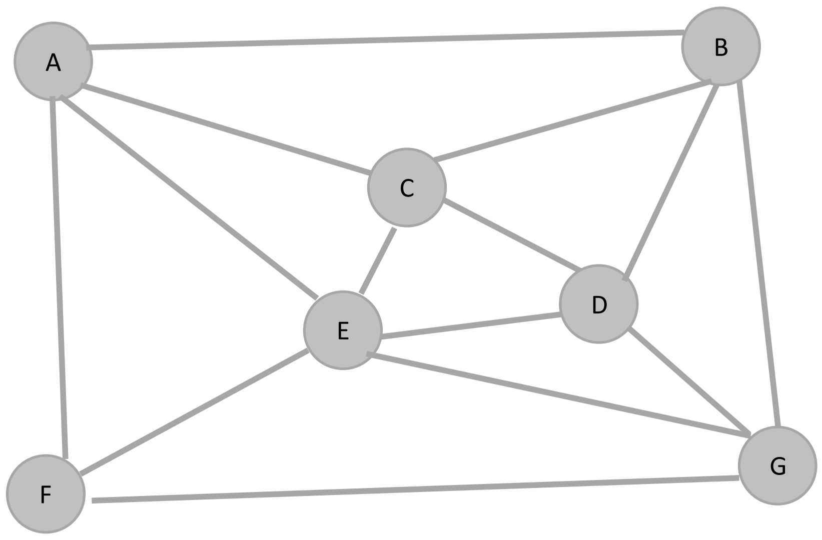

Fig.5

Question 6 on Next Page

Question 6: Heaps (Total Marls: 15)

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

A. Draw a heap using the number list, show you working as graphical steps, 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. **(4 marks)**

C. Show the process of tickle-up/tickle-down for inserting a value of "125" in the tree built in part A. Show steps on array-based representation built in part B. **(4 marks)**

D. How the heaps can be used for real-life application give an example of heap used in real-life. **(3 marks)**

13	131	51	45	28	93	102	80	74	66

Fig. 6

Question 7 on Next Page

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

A. Consider the following (Fig.7) has functions for implementation of double-hashing with a table length of 27 items. Use the hash function to insert the following value (Fig.8) in the hash table. Show your step including any collusions. **(6 marks)**

B. How you will manage a hash table to always stay at a maximum load factor of 0.6, what exactly the meaning of using this load factor in case of the hash table developed in Part A. Describe your answer by stating extra memory utalisation and frequency of resizing for this load factor selection. **(6 marks)**

C. A quadratic probing is doubling the prob length by power of 2 on each collusion in a hash table. This can cause an issue of secondary clustering. What particular property of a good hash function is missing in this technique. How it can be improved describe you comments by considering properties of a good hash function. **(6 marks)**

Hash function 1	Hash function 2

def hash1(key):
    hashIdx = (key >>1) % 3
    return hashIdx
	Def hash2(key):
	    hashIdx = key%2
	    return hashIdx

Fig. 7

Key	Value
24	“Apple”
31	“Orange”
1234	“Melon”
465	“Lemon”
789	“Carrot”
890	“Green Chilli”
93	“Plum”

Fig.8

Submission Guidelines on Next Page

2. Submission Guide Lines

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 files, name the code files appropriately e.g. "Q1PartA.py/java"

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.