**CS310 Data Structures**

**Final Exam**

**Total Points: 100 Number of Questions: 5**

**Name:**

**REDID:**

***Please read the questions thoroughly before answering.***

***Submit the exam as a PDF (written & scanned or filled in word and exported)***

1. **(20 Points) Time Complexities**

Assume the most time-efficient way to implement the operations listed below, assume no duplicate values and that you can implement the operation as a member function of the class – with access to the underlying data structure, including knowing the number of values currently stored (N).

• **Then, give the tightest possible upper bound for the worst-case running time for each operation in terms of N**, **Big-O**.

\*\*For any credit, **you must explain why it gets this worst-case running time** (No points will be awarded without explanation, even if Big O is correct)

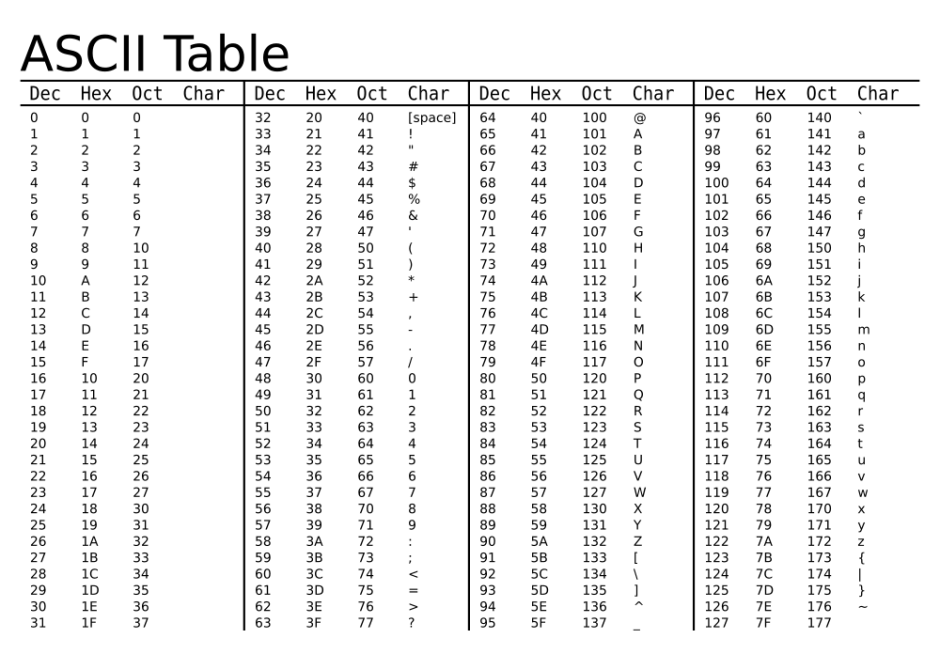
1. Sorting the elements of an array with Merge Sort
2. Given an input integer from the user, find if the integer is odd / even
3. Finding an element on a sorted array with Binary Search
4. Given a string input from the user (with n characters), find all permutations of the string.
5. Given a set of n numbers from the user, find all subsets of the set.

Example:

* Set is {1,2,3} Subsets are {},{1},{2},{3}, {1,2}, {1,3}, {2,3}
* {1,2} and {2,1} are considered the same.

1. **(20 Points) Hashing**

Given below is the ASCII Table for characters:



You are given the following set of keys to add to the hashtable

**a b e d**

**a b c d**

**d c b a**

**s e a x**

**c d e f**

**x a e s**

*Dictionary objects are (K,V) pairs. Here V Value does not matter. K keys is listed as above for the purpose of comparing hashfunctions.*

*The keys are 4-character Strings.*

Draco and Harry have each written a hashfuction to map the above keys into **a hashtable of size 10**.

**Draco’s Hashfunction**:

* + Find the **ASCII -> OCT** value of each character from the given table (*pay attention to the case, uppercase and lowercase are different OCT value*)
  + Add the remaining values and mod by 10 to get the index
  + In case of collision, use **Open Addressing with Linear Probing**. If you reach the end of the table loop back to top.

**Example:**

**Reading OCT values from the given table, all lowercase letters w,x,y,z**

**w x y z = 167+170+171+172 = 680 mod 10 = 0**

Using Draco’s function add the keys given at the start of the question to the following table and show the working for each key, *you must enter the values in the table and show the steps for each key in order to get any credit.*

|  |  |
| --- | --- |
| TABLE INDEX | KEY |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |

**Harry’s Function:**

* + Assign number to each character in the key based on their position in the Alphabet (A is 1, B is 2, C is 3 and so on)
  + Concatenate the first two numbers (join numbers do not add) and concatenate the next two numbers (join numbers do not add)
  + Add mathematically both concatenated numbers from previous step
  + Mod the number resulting from the previous step by 10 to get the index
  + In case of collision use **Open Addressing with Quadratic probing** (checking 1,4,9,16th position and so on). If you reach the end of the table, loop back to top.

Example:

**w is position 23, x is position 24, y is position 25, z is position 26**

**w x y z = SUM (“23”+”24” , “25”+”26”) = 2324 +2526 = 4850 mod 10 = 0**

Using Harry’s function add the keys given at the start of the question to the following table and show the working for each key, *you must enter the values in the table and show the steps for each key in order to get any credit.*

|  |  |
| --- | --- |
| TABLE INDEX | KEY |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |

**Extra Credit (2 Points) Based on the above results, whose hash function is better and why?**

1. **(20 Points) Data Structure applications**

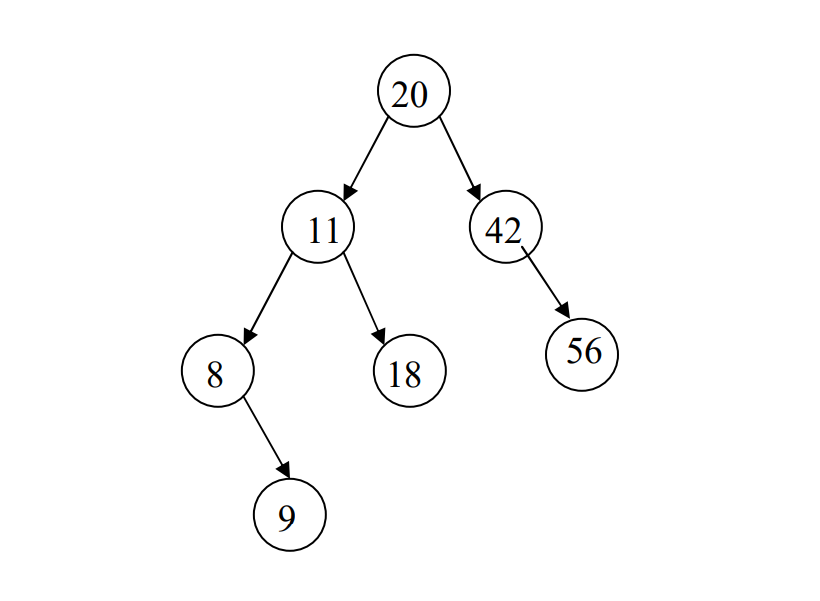
Answer the following questions based on the applications of Data structures.

1. In the project you are developing, adding each data must be a constant time operation. Data need not be maintained in any certain order. The amount of data grows dynamically. What data structure would you use and why?
2. You are writing a program to make a turtle walk through a maze. In order for the turtle to make the correct decisions, you must use backtracking to keep track of all the moves made before. What data structure would you use and why?
3. A Salesman is starting from his headquarters City A and has to visit 5 other cities. Given the distance between each of the cities and considering that all roads travel both ways (no direction to roads) , you need to find the shortest path that connects all the cities. Assume you are starting from City A and assume all cities are connected. What algorithm would you use and why?
4. My phonebook is now digitized and my database is sorted alphabetically. What algorithm should I use to find a person from the database and why?
5. **(20 Points) Graphs**

Given a weighted, undirected graph with V nodes, answer the following as best as possible, with a brief explanation. Assume all weights are non-negative.

1. If each edge has a unique weight, what can you say about the MST?
2. If the cost of an MST is ***c***, what can you say about the shortest distances returned by Dijkstra’s algorithm when run with an arbitrary vertex ***s*** as the source?
3. How many edges exist in the MST with V nodes?
4. If one vertex in the graph does not have any edges attached to it. What does that tell you about the Minimum Spanning Tree?
5. What algorithm would you use to find the MST if there are two edges between every given pair of vertices.
6. (1 Point Extra credit) – Other than Prim’s and Kruskal’s find one other algorithm to find Minimum Spanning Tree
7. **(20 Points Total) Trees**

Consider the following Tree:



1. (6 Points) Write traversals of the tree shown above:

Pre-Order:

Post-Order:

In-Order:

1. (6 Points) Consider the above Tree:

Circle yes or no to indicate whether the tree above might represent each of the following data structures.

**If you circle no, you must describe one way to modify the tree in order to make the answer into a yes.**

• AVL tree yes no

• Binary Search Tree yes no

• Full Binary Tree yes no

1. (8 Points) AVL Trees (Independent question not relating to above tree)

Draw the AVL tree that results from inserting the keys:

56, 78, 90, 23, 1, 4, 76, 34

**in that order into an initially empty AVL tree**.

**Showcase all intermediate steps and circle your final answer. Partial credit is awarded for correct intermediate trees.**