

CS201-Data Structures
Fall 2018
Assignment 4
Due: 29-NOV-2018
Section A, B, F: Submit on slate
Section C and D: Submit on Google classroom

Task 1:

Implement

- Function to find permutation of first n numbers.
- Double Hashing
- Quadratic Probing
- Random Probing
- Separate Chaining

Task 2:

Implement Heap Through linked List

Note: You have to implement all the functionalities of heap, which you have studied in the class.

Task3:

In this Task, you have to build chat application in which there are multiple users. All the users can communicate with each other through server via message sharing.

The Server should have heap, which prioritized the messages according to

- 1: First come first serve basis. (**Hint:** You can assign sequence number to the messages)
- 2: Message begin with "Emergency:" should have highest priority. (**Hint:** if there is more than one message starts with "Emergency" then priority should be set on basis of FCFS (first come first serve))
- 3: Message begin with "help:" should have second highest priority. (**Hint:** If there is, more than one message starts with "help:" then priority should be set on basis of FCFS (first come first serve))
- 4: Message with smaller length string have 3rd highest priority. (**Hint:** if there is more than one message with same length, starts with either "emergency:" or "help:" then priority should be set on basis of FCFS (first come first serve))

Note: Your program should be menu driven and should contain function to provide active users and function to display messages on heap.

Note: Your program should be implemented by linked list or Array (If you are using array-based implementation then it must increase its size dynamically.)

Task4:

Game of replacing array elements

There are two players A and B who are interested in playing a game of numbers. In each move a player pick two distinct number, let's say $a1$ and $a2$ and then replace all $a2$ by $a1$ or $a1$ by $a2$. They stop playing game if any one of them is unable to pick two number and the player who is unable to pick two distinct number in an array, loses the game. First player always move first and then second. Task is to find which player wins using hashing.

Examples:

Input : arr[] = { 1, 3, 3, 2, 2, 1 }

Output : Player 2 wins

Explanation:

First plays always loses irrespective

of the numbers chosen by him. For example,

say first player picks (1 & 3)

replace all 3 by 1

Now array Become { 1, 1, 1, 2, 2, 1 }

Then second player picks (1 2)

either he replace 1 by 2 or 2 by 1

Array Become { 1, 1, 1, 1, 1, 1 }

Now first player is not able to choose.

Input : arr[] = { 1, 2, 1, 2 }

Output : Player 1 wins

Task5:

Convert BST to Min Heap

Given a binary search tree which is also a complete binary tree. The problem is to convert the given BST into a Min Heap with the condition that all the values in the left subtree of a node should be less than all the values in the right subtree of the node. This condition is applied on all the nodes in the so converted Min Heap.

Task6:

Merge two binary Max Heaps

Given two binary max heaps as arrays, merge the given heaps.

Examples :

Input : a = {10, 5, 6, 2},

b = {12, 7, 9}

Output : {12, 10, 9, 2, 5, 7, 6}

Task7:

Count subarrays with same even and odd elements

Given an array of N integers, count number of even-odd subarrays. An even – odd subarray is a subarray that contains the same number of even as well as odd integers.

Implement it using hashing.

Examples :

Input : arr[] = {2, 5, 7, 8}

Output : 3

Explanation : There are total 3 even-odd subarrays.

1) {2, 5}

2) {7, 8}

3) {2, 5, 7, 8}

Task8:

We want to store the following data in a hash table with 11 size, in the order of their appearance.

25 96 42 223 112 12 84 102 153

Use division method for hash function and quadratic probing technique to resolve collision.

a. Draw and fill the hash table with the given data by using the above information.

b. How many comparisons are necessary to locate the record whose key value is 112?

c. How many comparisons are necessary to locate the record whose key value is 102?

d. How many comparisons are necessary to determine that the record whose key value is 14 is not in the table?

e. What happens if you remove the record whose key value is 223 from the table by just setting the field back to empty?

f. Draw and fill the hash table with the given data; by using rehashing to resolve collision. Use the last digit of your Roll# as a constant value in rehash function.