CSE 313: DESIGN AND ANALYSIS OF ALGORITHMS LAB

V SEM B.E.(CS &E)

(2012)

Prepared by

1. Mr. Ahamed Shafeeq B M
2. Mr. Gururaj

Approved by

(H.O.D)

DEPT. OF COMPUTER SCIENCE & ENGG.

M.I.T., MANIPAL

INSTRUCTIONS TO STUDENTS

1. Students should be regular and come prepared for the lab practice.

2. In case a student misses a class, it is his/her responsibility to complete that missed experiment(s).

3. Students should bring the observation book, lab journal and lab manual. Prescribed textbook and class notes can be kept ready for reference if required.

4. Once the experiment(s) get executed, they should show the results to the instructors and copy the same in their observation book.

5. The algorithms have to be implemented in C++.

6. Assume integer data, if explicitly not mentioned.

PROCEDURE FOR EVALUATION

The entire lab course consists of 100 marks. The marking scheme is as follows

Continuous Evaluation	60 marks
End Sem Lab Exam	40 marks
Total	100 marks

Scheme for continuous evaluation

Students will be evaluated bi-weekly. Minimum 6 evaluations should be conducted for each student. Each evaluation carries 10 marks. The scheme is as follows

Program and Execution	5 marks
Observation	3 marks
Viva-voce	2 marks
Total	**10 marks**

Scheme for end sem lab exam

End sem lab exam will be conducted after the completion of all the weekly exercises. The student will be not allowed for exam if he/she is found short of attendance and has not completed all the experiments. The marking scheme for end sem lab exam is as follows

Algorithm Design & Write-up of program	15 marks
Program execution	15 marks
Results for all inputs	10 marks
Total	**40 marks**

CONTENTS

WEEK 1

Review of fundamental data structures

1. Implement a doubly linked list to support the following operations

i. Create the list by adding each node at the front

ii. Insert a new node to the left of the node whose key value is read as an input.

iii. Delete all occurrences of a given key, if it is found. Otherwise, display appropriate message

iv. Search a node based on its key value

v. Display the contents of the list

2. Construct a binary search tree (BST) to support the following operations. Assume no duplicate elements while constructing the BST.

i. Given a key, perform a search in the BST. If the key is found then display “key found” else insert the key in the BST.

ii. Display the tree using inorder, preorder and post order traversal methods

Homework Exercises:

1. Repeat Problem 1 using singly linked list.

2. Construct a binary tree to support the following operations. You can assume duplicate elements while constructing the tree.

i. Insert a new node

ii. Delete a node based on its key value

iii. Display the tree using inorder, preorder and post order traversal methods

WEEK 2

Fundamentals of algorithmic problem solving

1. Implement GCD(greatest common divisor) using the following three algorithms.

   i. Euclid's algorithm

   ii. Consecutive integer checking algorithm

   iii. Middle school procedure which makes use of common prime factors. For finding list of primes implement sieve of Eratosthenes.

2. Find gcd(31415, 14142) by applying each of the above algorithms.

3. Find out which algorithm is faster for the above data. Estimate how many times it will be faster than the other two.

Homework Exercises:

1. Design an algorithm for computing $\lfloor \sqrt{n} \rfloor$ for any positive integer. Besides assignment and comparison, your algorithm may only use the four basic arithmetic operations.

2. Implement recursive solution to the Tower of Hanoi puzzle.

3. Compute the n^{th} Fibonacci number recursively.

WEEK 3

Brute Force techniques

1. Sort a given set of elements using bubble and selection sort and hence find the time required to sort elements.

2. Perform linear search and find the time required to search an element.

3. Given a string called TEXT with ‘n’ characters and another string called PATTERN with ‘m’ characters(m<=n) .Write a program which implements brute force string matching to search for a given pattern in the text. If the pattern is present then find the position of first occurrences of Pattern in that Text.

WEEK 4

Divide and Conquer

1. Sort given set of elements using

   -Merge Sort

   -Quick sort

and compare the time complexities of both the algorithms.

2. Perform binary serach

Homework Problems.:

1. Write a program to find mode using brute force strategy.(Mode: an item that occurs maximum number of times)

2.Write a program to find the height and count the number of nodes in a given tree.

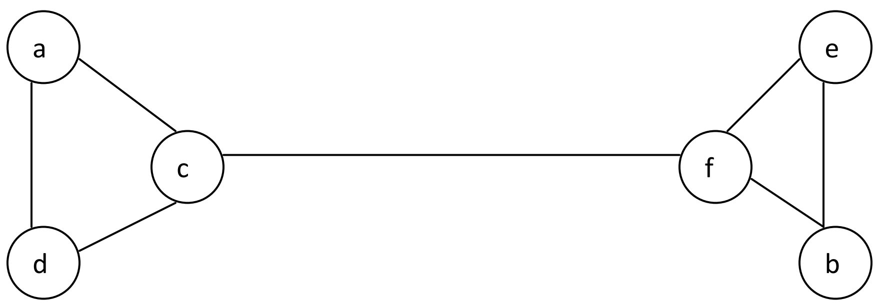
WEEK 5

Decrease and Conquer

1. Write an insertion sort program to sort the following data

   7 11 4 3 1 20 15 9 6 14

2. Wrtie a depth-first traversal algorithm for traversing the following graph starting at any node.



3. Write a breadth-first traversal algorithm for traversing the above graph starting at any node.

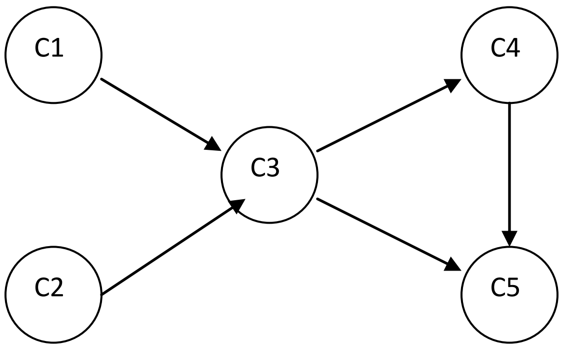
Homework exercises

1. Change the problem in exercise 1 above to binary search instead of sequential search when finding the position for insertion.

Week 6

Transform and Conquer

1. Write a topological sorting algorithm for the following directed graph.



2. Create a AVL tree for the following data.

   10 20 5 8 2 4 7 12

Homework exercises

1. For the AVL tree created in exercise 1 above, insert a element 6.
2. Delete a element 7 from the AVL tree in exercise 1.

WEEK 7

Transform and Conquer

1. Create a 2-3 tree for the following data.

9 5 8 3 2 4 7

3. Create a heap for the list 2, 9, 7, 6, 5, 8.

4. Sort the above heap using heap sort.

Homework exercises

1. For the 2-3 tree created in exercise 2 above, insert an element 6.
2. Delete a element 7 from the 2-3 tree in exercise 2.

WEEK 8

Space and Time tradeoffs

1. Implement Horspool algorithm for String Matching and find the number of key comparisons in successful search and unsuccessful search.

2. Construct the Open hash table. Find the number of key comparisons in successful search and unsuccessful search.

Homework exercises

1.Construct the closed hash table. Find the number of key comparisons in successful search and unsuccessful search.

2.Write a program to sort the elements using comparison counting and find

the time required to sort the elements.

WEEK 9 Dynamic Programming

1. Implement Floyd’s algorithm for the All-Pairs- Shortest-Paths problem(Graph –refer text book) .

2. Implement Warshall’s algorithm

Homework exercises

1. Find the Binomial Co-efficient using Dynamic Programming.

2. Compute the transitive closure of a given directed graph using Warshall's algorithm.(graph-refer text book).

WEEK 10

Greedy Technique

1. Find Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm.
2. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.
3. From a given vertex in a weighted connected graph, find shortest paths to other vertices Using Dijkstra's algorithm.

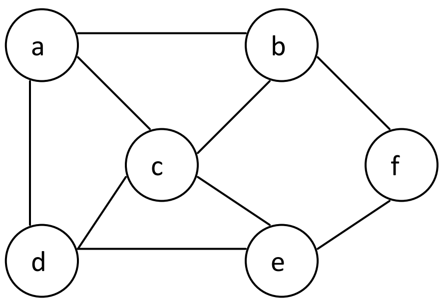
Homework exercises

1. Design an algorithm for finding a maximum spanning tree – a spanning tree with the largest possible edge weight of a weighted connected graph.
2. Implement Huffman tree construction algorithm.

WEEK 11

Backtracking

1. Write a program for N queens problem.
2. Write a program for finding Hamiltonian circuit for the following graph.



3. Find the solution to the subset-sum problem for S={1, 2, 5, 6, 8} and d=9.

WEEK 12

Branch and Bound

1. Implement Knapsack problem using branch and bound technique.

2. Implement assignment problem using Branch and Bound

Homework exercises

1. Implement Traveling Salesman Problem

WEEK 13

TEST

Test portion: WEEK 1 TO WEEK 12

References:

1. Anany Levitin, Introduction to The Design and Analysis of Algorithms, Pearson Education, 2nd Edition, 2007.
2. Horowitz E., Sahni S., Rajasekaran S.,Computer Algorithms by Galgotia Publications, 2001.

3. Thomas H. Cormen, Charles E. Leiserson, Ronal L, Rivest, Clifford Stein, Introduction to Algorithms, PHI, 2nd Edition, 2006.