CONTENTS

2.3.1 malloc() 28 2.3.2 calloc() 29 2.3.3 realloc() 29

Chapter 1. Introduction 1.1 Data Type 1 1.2 Abstract data types 1 1.3 Data structures 2 1.3.1 Linear and Non linear data structures 3 1.3.2 Static and dynamic data structures 3 1.4 Algorithms 4 1.4.1 Greedy algorithm 4 1.4.2 Divide and conquer algorithm 4 1.4.3 Backtracking 4 1.4.4 Randomized algorithms 4 1.5 Analysis of algorithms 4 1.5.1 Big O notation 6 1.5.1.1 Rules for O notation 6 Chapter 2. Arrays, Pointers and Structures 10 2.1 Arrays 10 2.1.1 One Dimensional Array 10 2.1.1.1 Declaration of 1-D Array 10 2.1.1.2 Accessing 1-D Array Elements 11 2.1.1.3 Processing 1-D Arrays 11 2.1.1.4 Initialization of 1-D Array 12 2.1.1.5 1-D Arrays and Functions 14 2.1.1.5.1 Passing Individual Array Elements to a Function 14 2.1.1.5.2 Passing whole 1-D Array to a Function 14 2.1.2 Two Dimensional Arrays 15 2.1.2.1 Declaration and Accessing Individual Elements of a 2-D array 15 2.1.2.2 Processing 2-D Arrays 15 2.1.2.3 Initialization of 2-D Arrays 16 2.2 Pointers 18 2.2.1 Declaration of a Pointer Variable 18 2.2.2 Assigning Address to a Pointer Variable 19 2.2.3 Dereferencing Pointer Variables 19 2.2.4 Pointer to Pointer 20 2.2.5 Pointers and One Dimensional Arrays 21 2.2.6 Pointers and Functions 23 2.2.7 Returning More Than One Value from a Function 24 2.2.8 Function Returning Pointer 25 2.2.9 Passing a 1-D Array to a Function 26 2.2.10 Array of Pointers 27 2.3 Dynamic Memory Allocation 27

2.3.4 free() 30
2.4 Structure 31
2.4.1 Defining a Structure 31
2.4.2 Declaring Structure Variables 32
2.4.2.1 With Structure Definition 32
2.4.2.2 Using Structure Tag 32
2.4.3 Initialization of Structure Variables 33
2.4.4 Accessing Members of a Structure 33
2.4.5 Assignment of Structure Variables 34
2.4.6 Array of Structures 34
2.4.7 Arrays within Structures 35
2.4.8 Nested Structures (Structure within Structure) 36
2.4.9 Pointers to Structures 38
2.4.10 Pointers within Structures 39
2.4.11 Structures and Functions 39
2.4.11.1 Passing Structure Members as Arguments 39
2.4.11.2 Passing Structure Veriable as Arguments 39
2.4.11.2 Passing Structure Variable as Argument 40
2.4.11.3 Passing Pointers to Structures as Arguments 40
2.4.11.4 Returning a Structure Variable from Function 41
2.4.11.5 Returning a Pointer to Structure from a Function 42
2.4.11.6 Passing Array of Structures as Argument 42
2.4.11.7 Self Referential Structures 43
Exercise 43
Chapter 3. Linked Lists 48
3.1 Single Linked list 48
3.1.1 Traversing a Single Linked List 51
3.1.2 Searching in a Single Linked List 53
3.1.3 Insertion in a Single Linked List 53
3.1.3.1 Insertion at the beginning of the list 53
3.1.3.2 Insertion in an empty list 54
3.1.3.3 Insertion at the end of the list 55
3.1.3.4 Insertion in between the list nodes 55
3.1.3.4 Insertion of terms of 55
3.1.3.4.1 Insertion after a node 56
3.1.3.4.2 Insertion before a node 57
3.1.3.4.3 Insertion at a given position 58
3.1.4 Creation of a Single Linked List 58
3.1.5 Deletion in a Single Linked List 59
3.1.5.1 Deletion of first node 59
3.1.5.2 Deletion of the only node 59
3.1.5.3 Deletion in between the list nodes 60
3.1.5.4 Deletion at the end of the list 60
3.1.6 Reversing a Single Linked List 61
3.2 Doubly linked list 63
3.2.1 Traversing a doubly linked List 65
2007
3.2.2. Insertion in a doubly linked Liet 45
3.2.2 Insertion in a doubly linked List 65 3.2.2.1 Insertion at the beginning of the list 65

Contents
3.2.3 Creation of List 69
2.2.4 Deletion from doubly linked list 69
3 2 4 1 Deletion of the first node 69
3.2.4.2 Deletion of the only node 70
3.2.4.3 Deletion in between the nodes 70
3.2.4.4 Deletion at the end of the list 70
3.2.5 Reversing a doubly linked list 72
3.3 Circular linked list 72
3.3.1 Traversal in circular linked list 74
3.3.2 Insertion in a circular Linked List 75
3.3.2.1 Insertion at the beginning of the list 75
3.3.2.2 Insertion in an empty list 75
3.3.2.3 Insertion at the end of the list 76
3.3.2.4 Insertion in between the nodes 76
3.3.3 Creation of circular linked list 77
3.3.4 Deletion in circular linked list 77
3.3.4.1 Deletion of the first node 77
3.3.4.2 Deletion of the only node 77
3.3.4.3 Deletion in between the nodes 78
3.3.4.4 Deletion at the end of the list 78
3.4 Linked List with Header Node 79
3.5 Sorted linked list 83
3.6 Sorting a Linked List 86
3.6.1 Selection Sort by exchanging data 87
3.6.2 Bubble Sort by exchanging data 88
3.6.3 Selection Sort by rearranging links 90
3.6.4 Bubble sort by rearranging links 92
3.7 Merging 93
3.8 Concatenation 97
3.9 Polynomial arithmetic with linked list 98
3.9.1 Creation of polynomial linked list 101
3.9.2 Addition of 2 polynomials 101
3.9.3 Multiplication of 2 polynomials 103
5.10 Comparison of Array lists and Linked lists 104
3.10.1 Advantages of linked lists 104
3.10.2 Disadvantages of linked liete 105
Exercise 105

Chapter 4. Stacks and Queues 108

4.1 Stack 108 4.1.1 Array Implementation of Stack 109 4.1.2 Linked List Implementation of Stack 111 4.2 Queue 114