

Analysis & Design of Algorithms (BCS401)

koku17

April 22, 2024

Contents

1	Module 1	1
1.1	INTRODUCTION	1
1.1.1	What is an Algorithm ?	1
1.1.2	Fundamentals of Algorithmic Problem Solving	1
1.2	FUNDAMENTALS OF THE ANALYSIS OF ALGORITHM EFFICIENCY	1
1.2.1	Analysis Framework	1
1.2.2	Asymptotic Notations and Basic Efficiency Classes	1
1.2.3	Mathematical Analysis of Non recursive Algorithms	1
1.2.4	Mathematical Analysis of Recursive Algorithms	1
1.3	BRUTE FORCE APPROACHES	1
1.3.1	Selection Sort and Bubble Sort	1
1.3.2	Sequential Search	1
1.3.3	Brute Force String Matching	1
2	Module 2	2
2.1	BRUTE FORCE APPROACHES	2
2.1.1	Exhaustive Search (Travelling Salesman problem and Knapsack Problem)	2
2.2	DECREASE – AND – CONQUER	2
2.2.1	Insertion Sort	2
2.2.2	Topological Sorting	2
2.3	DIVIDE AND CONQUER	2
2.3.1	Merge Sort	2
2.3.2	Quick Sort	2
2.3.3	Binary Tree Traversals	2
2.3.4	Multiplication of Large Integers and Strassen’s Matrix Multiplication	2
3	Module 3	3
3.1	TRANSFORM – AND – CONQUER	3
3.1.1	Balanced Search Trees	3
3.1.2	Heaps and Heapsort	3
3.2	SPACE – TIME TRADEOFFS	3
3.2.1	Sorting by Counting	3
3.2.2	Comparison counting sort	3
3.2.3	Input Enhancement in String Matching (Horspool’s Algorithm)	3
4	Module 4	4
4.1	DYNAMIC PROGRAMMING	4
4.1.1	Three basic examples	4
4.1.2	The Knapsack Problem and Memory Functions	4
4.1.3	Warshall’s and Floyd’s Algorithms	4
4.2	THE GREEDY METHOD	4
4.2.1	Prim’s Algorithm	4
4.2.2	Kruskal’s Algorithm	4
4.2.3	Dijkstra’s Algorithm	4
4.2.4	Huffman Trees and Codes	4
5	Module 5	5
5.0.1	LIMITATIONS OF ALGORITHMIC POWER	5
5.0.2	Decision Trees	5
5.0.3	P, NP, and NP – Complete Problems	5
5.0.4	COPING WITH LIMITATIONS OF ALGORITHMIC POWER	5
5.0.5	Backtracking (n – Queens problem, Subset – sum problem)	5
5.0.6	Branch – and – Bound (Knapsack problem)	5
5.0.7	Approximation algorithms for NP – Hard problems (Knapsack problem)	5

1 Module 1

1.1 INTRODUCTION

1.1.1 What is an Algorithm ?

1.1.2 Fundamentals of Algorithmic Problem Solving

1.2 FUNDAMENTALS OF THE ANALYSIS OF ALGORITHM EFFICIENCY

1.2.1 Analysis Framework

1.2.2 Asymptotic Notations and Basic Efficiency Classes

1.2.3 Mathematical Analysis of Non recursive Algorithms

1.2.4 Mathematical Analysis of Recursive Algorithms

1.3 BRUTE FORCE APPROACHES

1.3.1 Selection Sort and Bubble Sort

1.3.2 Sequential Search

1.3.3 Brute Force String Matching

2 Module 2

2.1 BRUTE FORCE APPROACHES

2.1.1 Exhaustive Search (Travelling Salesman problem and Knapsack Problem)

2.2 DECREASE – AND – CONQUER

2.2.1 Insertion Sort

2.2.2 Topological Sorting

2.3 DIVIDE AND CONQUER

2.3.1 Merge Sort

2.3.2 Quick Sort

2.3.3 Binary Tree Traversals

2.3.4 Multiplication of Large Integers and Strassen's Matrix Multiplication

3 Module 3

3.1 TRANSFORM – AND – CONQUER

3.1.1 Balanced Search Trees

3.1.2 Heaps and Heapsort

3.2 SPACE – TIME TRADEOFFS

3.2.1 Sorting by Counting

3.2.2 Comparison counting sort

3.2.3 Input Enhancement in String Matching (Horspool's Algorithm)

4 Module 4

4.1 DYNAMIC PROGRAMMING

4.1.1 Three basic examples

4.1.2 The Knapsack Problem and Memory Functions

4.1.3 Warshall's and Floyd's Algorithms

4.2 THE GREEDY METHOD

4.2.1 Prim's Algorithm

4.2.2 Kruskal's Algorithm

4.2.3 Dijkstra's Algorithm

4.2.4 Huffman Trees and Codes

5 Module 5

5.0.1 LIMITATIONS OF ALGORITHMIC POWER

5.0.2 Decision Trees

5.0.3 P, NP, and NP – Complete Problems

5.0.4 COPING WITH LIMITATIONS OF ALGORITHMIC POWER

5.0.5 Backtracking (n – Queens problem, Subset – sum problem)

5.0.6 Branch – and – Bound (Knapsack problem)

5.0.7 Approximation algorithms for NP – Hard problems (Knapsack problem)