

Guru Gobind Singh College of Engineering and Research Centre, Nashik



Expe	rimer	nt No	o: 01

Write a program non-recursive and recursive program to calculate Fibonacci numbers and analyse their time and space complexity.

Student Name:						
Class:	BE (Compute	er)				
Div:				Batch:		
Roll No.:						
Date of Attendance (Performance):						
Date of Evaluation:						
Marks (Grade)	A	P		W	Γ	Total
Attainment of CO						
Marks out of 10						
CO Mapped	_	nt an algorithm	that i	gorithm. follows one of ter, greedy, dyna		
Signature of Subject Teacher						



Experiment No: 02

Guru Gobind Singh Foundation's



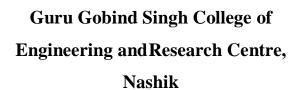


-			

Write a program to solve a fractional Knapsack problem using a greedy method.

Student Name:							
Class:	BE (Comput	ter)		<u>, </u>			
Div:			Batch:				
Roll No.:							
Date of Attendance							
(Performance):							
Date of Evaluation:							
Marks (Grade)	A	P	W	T	Total		
Attainment of CO							
Marks out of 10							
CO Mapped							
	CO4: Analyze	performance	of an algorithm	1.			
	CO5: Implement an algorithm that follows one of the following						
	algorithm design strategies: divide and conquer, greedy, dynamic						
	programming, backtracking, branch and bound.						
Signature of							
Subject Teacher							







Experiment No: 03

Write a program to solve a 0-1 Knapsack problem using dynamic programming or branch and bound strategy.

Student Name:							
Class:	BE (Compu	ter)					
Div:			Batch:				
Roll No.:			,	,			
Date of Attendance							
(Performance):							
Date of Evaluation:							
	A P W T Total						
Marks (Grade)							
Attainment of CO							
Marks out of 10							
CO Mapped	CO4: Analyse	performance of	of an algorithm	1.			
	CO5: Impleme	ent an algorith	m that follows	one of the fol	lowing		
	algorithm desi	ign strategies:	divide and con-	quer, greedy,	dynamic		
	programming,	, backtracking,	branch and bo	und			
Signature of							
Subject Teacher							
			<u></u>				



Guru Gobind Singh College of Engineering and Research Centre, Nashik

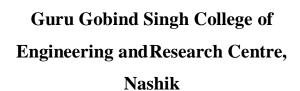


Experiment No: 04

Design n-Queens matrix having first Queen placed. Use backtracking to place remaining Queens to generate the final n-queen's matrix.

	1						
Student Name:							
Class:	BE (Comput	ter)					
Div:			Batch:				
Roll No.:							
Date of Attendance							
(Performance):							
Date of Evaluation:							
Marks (Grade)	A	P	W	T	Total		
Attainment of CO							
Marks out of 10							
CO Mapped		C	C 1 '41				
	CO4: Analyse performance of an algorithm.						
	CO5: Implement an algorithm that follows one of the following						
	algorithm design strategies: divide and conquer, greedy, dynamic						
	programming,	backtracking, l	oranch and b	ound.			
Signature of							
Subject Teacher							







Experiment No: 05

Write a program for analysis of quick sort by using deterministic and randomized variant.

Student Name: Class: BE (Computer) Div: Batch: Roll No.: Date of Attendance (Performance): Date of Evaluation: Marks (Grade) Attainment of CO Marks out of 10 CO Mapped CO4: Analyse performance of an algorithm. CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound. Signature of Subject Teacher						
Div: Roll No.: Date of Attendance (Performance): Date of Evaluation: Marks (Grade) Attainment of CO Marks out of 10 CO Mapped CO4: Analyse performance of an algorithm. CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound. Signature of	Student Name:					
Roll No.: Date of Attendance (Performance): Date of Evaluation: Marks (Grade) Attainment of CO Marks out of 10 CO Mapped CO4: Analyse performance of an algorithm. CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound. Signature of	Class:	BE (Compute	er)			
Date of Attendance (Performance): Date of Evaluation: Marks (Grade) Attainment of CO Marks out of 10 CO Mapped CO4: Analyse performance of an algorithm. CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound. Signature of	Div:			Batch:		
(Performance): Date of Evaluation: Marks (Grade) Attainment of CO Marks out of 10 CO Mapped CO4: Analyse performance of an algorithm. CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound. Signature of	Roll No.:					
Date of Evaluation: Marks (Grade) Attainment of CO Marks out of 10 CO Mapped CO4: Analyse performance of an algorithm. CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound. Signature of	Date of Attendance					
Marks (Grade) Attainment of CO Marks out of 10 CO Mapped CO4: Analyse performance of an algorithm. CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound. Signature of	(Performance):					
Marks (Grade) Attainment of CO Marks out of 10 CO Mapped CO4: Analyse performance of an algorithm. CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound. Signature of	Date of Evaluation:					
CO Mapped CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound. Signature of	Marks (Grade)	A	P	W	T	Total
CO Mapped CO4: Analyse performance of an algorithm. CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound. Signature of	Attainment of CO					
CO4: Analyse performance of an algorithm. CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound. Signature of	Marks out of 10					
	CO Mapped	CO5: Implement design strategies:	an algorithm	that follows one onquer, greedy, c	of the following	
Subject Teacher	Signature of					
	Subject Teacher					







Experiment No: 06

Mini Project - Write a program to implement matrix multiplication. Also implement multithreaded matrix multiplication with either one thread per row or one thread per cell. Analyze and compare their performance.

Student Name:							
Class:	BE (Computer)						
Div:			Batch:				
Roll No.:							
Date of Attendance							
(Performance):							
Date of Evaluation:			T	T	T		
Marks (Grade)	A	P	W	Т	Total		
Attainment of CO							
Marks out of 10							
	CO4: Analyze performance of an algorithm. CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound.						
Signature of							
Subject Teacher							