	Shreeya Thigale TE-IX
	TE-IX
	33164. PAGE No. DATE //
=	
	-ASSIGNMENT-13
	117LE: Travelling Salesman Problem (BB)
	PROBLEM STATEMENT:
	Write a program to solve the travelling salesman problem and to print the path and cost using
	problem and to print the path and cost using
	Branch and Bound.
	OBIECTIVE:
	To renderstand and implement Least Count
	OBTECTIVE: To renderstand and implement Least local Branch and Bound algorithm for solving Travelling Salesmen problem and study Branch and Branch Strategy.
	Salismen problem and study Branch and
	Bound strategy.
	THEORY:
	Branch and Bound Strategy
3.	Branch and bound (BB for B&B) is an
ğ, s	algorithm disign paradigm for discrete and
	1 1 f-3 1 continues at the months, we will
	general real valued problems in strains
	hound algorithm consists of systemic ensensation
	1 00 01 10 10
	Search's the set of cardidate solutions is Thought of as forming a rooted tree with full set at root.
	thought of as forming a morrie 1.2
	full set at root.
mitter.	4000

													1	PAGE	No.	//	7
			,	1	,	1	1				1)			
7											1						
	Least lost Branch and Bound																
														/	1	1-	
	((In	0	rdu	. 1	D	w	e_	1C	br	anch	<u></u>	1. 1-	nd_	100	
	In order to use 10 branch bound to Search the travelling subspecsor state space true, we need to define a cost function (1) and two other functions (1) and u(1) such that (2(R) < C(R) < u(R) for all																
	true we need to define a cost function																
٥		cc)	an	d	t	W)	oth	n 1	June	ction	<u>v</u>	0)		an	d
	u() such that $\hat{c}(R) \leq c(R) \leq u(R)$ for all																
	n	odu) A	2											<u> </u>	- (-	
		2															
	C	xa	mple	, ,													
	=								7 N]			10	17	· a	1 1
23 - 1 4		0		20		30		<u>(0</u>		11 2		C0	2_	10 CS	1)	2	
		+ '	5	5		16	1.4	2		4	7		0	3	00		
		10		6		12		00		3		15		3	12		
-		+-	6	4		7		16		80		}		٥	O	12	124 12
				<u>_</u>	art.	ma							R.	duc	ed (pst	
					000	7140	v	/ <u>X</u>						meet			
	7	00	00	cs	co	05	'	00	Ø	os	00	0		B	cs	Cs	00 00
		Ø	00	11	2	0		1	Ø	00	2	U		12	Co	11	00
3 (7	0	Cs	00	Ø	2		00	3	O	0	2		0	3	a co	0 2
		15	O	12	0	0		4	3	00	00	0		00	3	12	00 0
4	Ä.	11	00	0	12	05		6	0	∞	12	∞		11			00
<u> </u>		a)pa	th 1,	2', no	dis	2		<i>b</i>) <i>p</i>	ath	1,3;	nocle	3		c)	path	2 174	i) node4
	\vdash			^		7		-	1.			-					
		0	Ø	9	0	cs cs	r	00	09	0	Øs.	∞		00	00	11/2	00 00
	,	Q	3	00	0	05		0	00	11	05	0	_	J.	O	00	000
	ordis a si	12	0	9	Ø	0	17	8	00	00	00	2		00	0)	0	10
		00	0	0	12	00		11	00	0	0			0	0	CARRY ST	00
	0	() p	ian)	5, 4	rocl	15		e)	pat	11,4	2;	and .			,,		3 nodel
									1	6		- Land		8	path	11-11-	

PAGE No.		
DATE	//	

1							/ PAGE No. /
-							DATE / /
7			,)	
+		Co	co.	05	00	00	000000000000000000000000000000000000000
+		1	CO	0	CS	cs	000000000000
1		0	3	00	00	00	000000000000
1		05	cs	CS	00	0	
-		0	٥	0	O	0	000000000000000000000000000000000000000
-		a)	pall	1,4,5	inoc	48	h) path 1,4,2,3; i) path 1,4,2,5;
		01	<i>j</i> -cc <i>i</i> -) //	, , , ,		node 9 node 10
_							
_							(1) 25
_				i,= 2			
_		35 (2)		i,	= 3/	i,=4 i,=3 (5)3
_				5.	3 (2	5	4)25
							1:2
							i,:3
						28	(6) 50 (7) (8) 36
						2 /	i ₃ -5
					î3=		
				52	9		10) 28
							<i>i</i> μ=3

State space tree generated by procedur

11) 28

FOR TSP using LCBB mithodis int cost; int path [10]

	PAGE No.	
	/ DATE /	11
	int "matrix [10][10];	
	int matrix (10) (13)	1
	3; typedej smeet lin hterm;	
	typedel smeet lim tour	
-		
-	hkum node, list [25];	
	1 (3.36.1 [10])	
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	. 4
-	Int red-mat [10][10], temp [10][10] 3. Read the no. of cities n and read the	C 15 p-101
-	+ 3	
	a Midiza god-matrix to kep-cost matrix	
	3. cost = reduce matrix (tsp-cost-matrix)	
	4. node. cost = cost	
	node. path [0] = 1;	
_	node. path[1]=1)	
	noch matrix = reduced matrix	
	5. node = expand (node);	
	6. if node cost < list[1]. cost	
	goto step 14	
	7. else	
	8. while (1) do	
	9. if size of heap is 0 break;	
	10. node = delete();	
	11. node = expand (hode);	
	if (node.cost < list [1].cost)	
	13. end do	
	14. Print path using node path	
	at nocle	
	16. you can verily that	2
as as the same	cost matrix as well gine	
	17. stop.	

PAGE No.
function expansion (node)
1 1:1. (1) do
1. while (1) do
2. do 3. count = node path [0];
3. count = node path (OS) 4. k = count +1;
c de monte conti
6. Store node matrix to some temperory matrix say
temp-matrix
7. 9= node. path Tevent)
2 1 1-1 ha a set visited (i) = 0)
9. for i=1 to count set visited [path [i]] = 1;
10. for j=2 to n
11. Begin for
12, 4) (Nource Ep)
13. Begin if
14. copy the temp-matrix to red matrix
13. See 11/10/19 + (1 + i):
1 t= cost + cost + temp-matix [i][j]
1 4 507 - Ki
1 1 0 1 7 2 1 .
1' reduced make & oftweet in sign
21. Insut (node)
22. end if
22 end for
24. 41 (K=Fn) break)
25. node = Hecketi()
26. End while
29. Robure node.

The second second	DATE
	Analysis?
	Thatyste TSP complexity will not be while worst case TSP complexity will not be any better than $b(n^22^n)$ the use of good any better than $b(n^22^n)$ the use of good bounding functions will enable these branch bounding functions will enable some problem
	any better than will enable these branch
	and bound algorithms to solve some problems and bound algorithms to solve some problems
	and bound algorithms than required by instances in much less time than required by
	instances in much ses algorithm. dynamic programming algorithm.
	CONCLUSION:
	Thus we have studied Least Cost Brand
	and Bound strategy for 7SP and also implemented it successfully.
	implemented it successfully.