	Yatish Kelkar TE 17 80
To Holl	DAA Assignment G
add ()	of the two nodes are no a spanning as AAD Implementation of the nodes as a queue, least a remarks head of queue and
	adde noduto top of stack
*	Aim: Write a program to solve TSP & to print path and cost usir
	branch & bound again and bound and only
(3000)	nodes are userted in a stack legst() paps from stock ?  pushes anto stack.
*	
1)	What is branch & bound?
Ans	Branch & bound is an algorithm design paradigm for discrete &
	combinatoral optimization. It consists of systematic enumeration
	combinatorial optimization. It consists of systematic enumeration canditate solutions by means of state space search.
ĨI)	The set of condidate solutions is thought of as forming a rooted tree u
	set of roots. Algorithm explores branches of this tree which represent
	subsets of solution set or doorgo paramoner and
iii)	sibproblems Foch subproblem takes a time so total tir
111)	Before enumerating canditate solutions of a branch, it is also check against upper & lower estimated bounds on the optimal solution is discarded if it cannot produce a better solution than best on
	is discarded if it cannot produce a better solution than best on
	found by algorithmenesles pulls and soul no subsol
	bot constantly undirected & implemented
2)	Explain FIFO UFO & LC hound & hound.
Ans:	Explain FIFO, UFO & LC branch & bound.  i) FIFO bronch and bound: It is a breadth first search. Ch
Sundaram	FOR EDUCATIONAL USE

Priev ii)	of the live nodes are inserted in the queue. Implementation of list live nodes as a queue, least () removes head of queue and add ( adds node to top of stack.  LIFO branch & bound: It is a depth first search. Children of live nodes are inserted in a stack. least () pops from stock & add ( pushes onto stack.
	LC branch & bound: The sclection rule for next live node in the branch FIFO or LIFO and linked is sometimes blind, ie no preference is given to a node that has a good chance of quickly finding a response node. It can be speeded up by using Intelligent ranking function or approximate cost function.
chacked	Dynamic programming approach breaks the problem into 2 <sup>n</sup> subproblems. Each subproblem takes n time so total time complexity is $o(2^n n^2)$
	Conclusion: Thus the trovelling sales man problem has been successfully understood & implemented.
Gundaram	based & dead of A only one heart and one of the dead o