Section &

Q3 Algorithm to compute Transitive & Clasure

for (i=0 to n-1) do

for (j=0 to n-1) do

P[i][j] = A[i][j]

end for

end for

for (k=0 to n-1) dofor (j=0 to n-1) dofor (j=0 to n-1) doif $(P[i][j]=0 \text{ and } P[i,k]=1 & & \\ P[k,j]=1$ then

T= [[:]]4

end tor end tor

Time efficiency of warshall's Alganismm $f(n) = \sum_{k=0}^{n-1} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} 1$ $= \sum_{k=0}^{n-1} \sum_{i=0}^{n-1} (n-k-0+k)$

$$= \sum_{k=0}^{n-1} n_{x} \sum_{i=0}^{n-1} 1$$

$$= \sum_{k=0}^{n-1} n_{x} (n-1-0+1)$$

$$= \sum_{k=0}^{n-1} n^{2}$$

$$= n^{2} \sum_{k=0}^{n-1} 1$$

$$= n^{2} (n-1-0+1)$$

$$= n^{3}$$
So, Time camplewity = $0(n^{3})$

04

Prin's Program

include < stdio.h>
include < conio.h>
define INFINITY 999

int prim(int cast[10][10], int source, int n)

int i, j, visited[10], verten[10], cmp[10];
int min, u, v, sum=0;
for[i=1; i<=n; i+t)
}

verten [i] = source;

```
visited[i]=0;
   cmp[i] = cost [source ][i];
  visited Esource ] = 1;
  for (i=1; i<=no-1; i++)
       min = INFINITY;
       for (j=1 ; j <= n; j++)
       if [! visited[j] && cmp[j] < min)
          min = cmp[j];
           u= j3
       visited [u]=1;
        sum + = cmp[u];
        print+ (" \n. 7. d -> 1. d sum = 1. d",
                 verten [u], u, som);
         for ( = 1; 0 <= n; 0++)
         @ if [! visited[v] && cost[u][v] < cmp[v])
             :[v][u]tas = [v][v];
             verten [v] = u;
           3 no stepp show you will
         The Agranages James Second
void main ()
```

in alsollsol, n,i,j, m, source, k;

printf(" Enter no. af vertices");

sconf("v.d", &n);

printf("Enter cast motrin);

tor li=1; i<=n; i+t)

for (j=1; j<=n;j+t)

sconf("v.d", & ali)[j];

printf("Enter the source");

sconf("v.d", & source);

m = prim(a, source, n);

printf("cost = v.d", m);

getch();

3

Section B LECTION

(muz en La Justia)

Divide & Conquer Method

- (1) It deals three steps at each devel of secursion: Divide, Conquer, Combine.
 - (2) It does more work on subproblems a nence has more time consumption.
- (3) It is a topdown approach.
- (4) In this subproblem are independent of each other,
- (5) It is Recursive.
- (6) Example: Merge Sort

Dynamic Programing

STATE OF THE PARTY	
(1)	It involves sequence of four steps. (i) Characterize the structure.
	(ii) Recursively define the values
401 3	(iii) compute the value.
34	(iv) Construct the optimal solution.
(2)	It solves subproblems only once & then stores in the table.
(3)	It is a Bottom - up approach.
(4)	In this subproblems are solved dependently.
(5)	It is non recursive.
(1	Evample: Fihnarci Series.

		1	_	
02	Bionomial	Cotticient	at	6 C4
9-	9.0		-	-

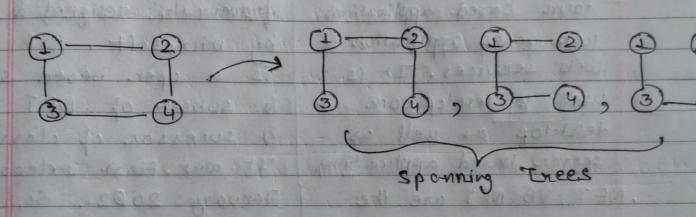
			- Aller Edward				-
THE REAL PROPERTY.	THE WARREN	THE PARTY	. 0	+	2	3	4
C(6,4) = C(5,3)	3) + c(5,4)	10	1	1///	11/1	1111	1111
c(5,3) = c(4)		1	1	1	11/1/	1/11	1/1
C(5,4) = C(4,		2	1	2	1	1111/	111
C(4,2) = C(3)		10 13	T	2	2	T	
C(4,3) = C(3)		0 4 n 4	1	2	?	5	1
C(3,1) = C(3,1)		5	1	?	?	?	2
C[3,2] = C(4006	1] ?	5	5	?
	(1,0) + (1,1)		1	HUGHL.	1		

		NE TO		_		1<	
				0 1	2	3	4
So,			. 01	1 1///	11/1/	11111	1111
	1) = 1 + 1 :	= 2	1	1 1	1////	(111)	1/11
c (3	(2) = 2 + 1 :	= 311 14 16	2	1 2		1/1/1	11/1
CC	$3_1 \perp) = 1 + 2$	=31	3	1 3	3	1	1///
	4,3) = 3+1		11	1 5	6	4	1
	(4,2) = 3+3		5	Taling	27 18:3	10	5
	(5,4) = 4 A		6	7		P	15
	(5,3) = 6 +		1		De no	15	4.3
C	(6,4) = 10	The second second second			TOTAL I		
	TAU TA		noital \		77		
THINSTIP	So,	C(6,4) =	15	334133		No.	Selection of the last of the l
		TORE STATE		1		USI, TE	S119
	13 (18) 10		TO BE	Man alma	NOT PER	1	
	epollianies la	Section A	Yes				
	Make - make t	16.12 Jan 19.38 3 19	SVohille	3 bino	rais .	2	2
Or	Transitive		Garage.				
1		A STATE OF THE STA			Neve	N.S.	10000
313.4		there is a					
	then the	re exist a	path	from ((iei)	AEM	100
Enemp	le: Input:	1 707	The state of the s	TOTAL	i la la la		10
		0110	(1/2)	0-		(2)	
	1 30 2	0011	10 000	11	1	(T)	
C The	a chard	0001	C 100		/	127	30
No.	Output :	1 1 1 1 1	LOLD W	VIV		00 T	
Walter St.		0 + 1 1		(2)_		(3)	
FILL	A CONTRACTOR	0011	S. Lekui				
S. S.		0001					

92

Sponming Tree

Sponning Tree is a tree where all nodes should be connected & graph should be dacyclic. delle golevol soull be a letter



Optimal Solution

An optimal solution is the solution where the objective function reaches the maximum minimum value.

For enample: The most profit as the deast 434 John of the west cost.

Two functions of Kruskal's Algarithm (1) find () (2) union ()

Problems that can be solved using dynamic:

(4) Position problem

Matrin Chain Multiplication

coin change Problem (3)

word break Problem 141