



COURSE NAME: ANALYSIS OF ALGORITHM II YEAR/ IV SEMESTER UNIT – III

DYNAMIC PROGRAMMING

Topic

Dynamic Programming: Binomial Coefficient

SNS COLLEGE OF ENGINEERING





Coimbatore-107

	Unit in <u>Dynammic</u> Page
	Dynammic programming is a mother
	in which the solution is a
	problem is obtained by making
	Sequence of decisions.
	Sequence of decisions. * Optimal Solution is obtain.
	for sequence of all possible
	Solutions generated.
	A TA or griven ay Uso Matherice
	Richard Bellman in 1950,
24.20	or it solves problems with
	over lapping sub peoblems.
100	
	(i). Computing Binomial coefficient:
Carl In St	To combinatorics, Binomial
	coefficient is a coefficient
	of any of the terms in
	the esepansion of (a+b)n,
	internet and the second s
	A. It is denoted by C(h, K)
	A. It is denoted by c(n, k) ar (n) where (D≤K≤n)
	Formula for Computing Binomial Cottin
Pascal I deut	
- Levil	y = c(n-1, k-1) + c(n-1, k)
100	y C(n,k) = c(n-1,k-1) + c(n-1,k) and $e(n,0) = 1$, $c(n,n) = 1$; $n > k > 1$



SNS COLLEGE OF ENGINEERING

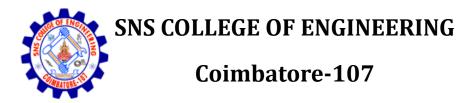


Coimbatore-107

S. Martin	Pol Dork K classmate
	(a+b) = = = C(n,k)a ^{n-k} , K classmate Example: Compute C(4,2) Date
_	Example: Compute (4,2)
	NET KEZA
-	(i) compute ((4,2) = c(n-1, k-1) + (c(n-1), k)
	$\frac{C(4,2) = c(4-1,2-1) + c(4+1), k}{c(4+1), k}$
	z = (2, 2) = c (3, 1) + c (3, 2)
	Both c(3/1) & c(3,2) are unknown. So compute it, to solve Equal.
	So compute it, to solve Equal. p(D=3, k=1) Gi) compute $c(3_{11}) = c(n-i, k-1) + c(n-1), k$
	(i) compute c(3,1) = c(n-i, k-1) + c (n-1), k
	= c(3-1, 1-1) + c(3-1), I-
	= c(2,0) + c(2,1)
	(ii) compute c(3,2)= c(n-1, k-1) + c(n-1, k)
	= c(3-1,2-1) + c(3-1,2)
	= c(2,1) + c(2,2)
	We should solve c(2,0), c(2,1),
	C(2,2) to some Equer D
	(ii) compute $C(2,0)$ and $C(2,2)$ $C(2,2)$ From formula: $C(D,D)=1$; $C(D,D)=1$
	(iv) compute $C(a, 1) = C(n-1, k-1) + C(n-1, k)$
	= c(3-1:t-t) + c(2-1:t)
	=c(1, 0)+c(n, 1)

SNS COLLEGE OF ENGINEERING Coimbatore-107 www.snsgroups.com Know 210 non C (1,0) + 1, C C • * C(es 2 2 r +C 2% 2 .D C2 1) C 1 2 vo 3 Va C 9 au n C 3 4,2

Unit 3/II CST/Analysis of Algorithm/K.Priyanka/AP/CST/SNSCE





compute c(n,k), Smaller overlapping To Sequencis generated by) c(n-1, K-1) & c(n-1, -) Smaller Instances Solved =) These Solutions generate Solution Algorithm: Algorithm Binomial (n,K) ton ton) 20 00 to t) do 20 (00) (i== j)) then eli,J= else [i, j]= C[i-1,j-1] + C[i-1,j' C neturn c[n,k] Analysis: <u>Time complexity</u> F(n)= O(nk) (n+1) * (k+1) table Since we fill Space Complexity! scn) = o(nk).100 Mar 100

Unit 3/II CST/Analysis of Algorithm/K.Priyanka/AP/CST/SNSCE