	Dynamic Programming	
	the commence of the Commence o	The last to the la
	1) Greedy Method -> decision	is taken one time
	2) Dynamic Programming -> decision	on taken at each step
	A second second	a late to be to
	principle of optimality	
	The property from the second second	of promoth to be night
ex'	Y The second sec	for K from I be
10.00	$fib(n) = \begin{cases} 0 \\ fib(n-2) + fib(n-1) & if n > 1 \end{cases}$	0,1,1,2,3,5,8,13,21
	tib (n) = (fib(n-2) + fib(n-1) if n>1	fib(5)
	inf fib (int n) hear la	
	ξ	+ fib(4) fib(3)
	if (n <= 1)	fib(3) fib(2)
	return in;	fib(1) + fib(2) fib(0) fib(1) fib(0) from
	return fib (n-2)+fib(n-1);	fib(o) fib(i)
	3 - (1911) X 2 < 0	
		T(N) = 2T(N-1)+1
	TOTAL LANGE TOTAL CONTRACTOR	(0(2 ^N)
27%	Reduce function calls of fb(1)	£ C 12
	getting called increasing our b	oig O.
	. and white the statement of the	CALL TO THE PROPERTY OF THE PR

	the Sparse Management and the second
	F -0 -1 -1 -1
	2 3 9 5 1 0 1
	The the Samuel of a 2 month with a first 12 to
	Start out at (-1) for all keep going through the
	fib(N) graph solving all of these to save time complexes
-3	when Solving!
-	to come the one yours come
-	F 0 1 116 2 (1 10) 3 + (5-111)
	0 12 3 4 5
	De Sula - Cravalla (a sai) and sai
	memoization has made it from o(2") to u(N), 15 calls
	to just 100.6. Man in the state of the state
	the man of the company
	memoization follows top down approach.
	S S CO Sand Street Stre
	9 3
2	0 14 CE-03+ EE-03+ EM 1
	C 12 John Stores to Chemin
	D 16 CN73 mora
	6 10 (3) 8 h B - 2 B = 21/2 + 4 = 3 (hour)
	571 3 (3) (1) (2) (2) (3)
	[[Elver 1] 5 lde Noto for Pyramic Personal
9	(1) 5 7 A 7 C 7 C = 7 4 5 6 7 7 4 7 7 A
-	BOTH THE STATE OF PRINCIPLE OF ME WELL IN BUILD
	Lower Thomas The Company

Tabulation	
for the state of t	
F 0 1 1 2 3	5
20 1 2 3 4 5	
$fib(n) = \begin{cases} 1 & \text{if } n = 1 \end{cases}$	" priving reduce
fib (n-2) + fib (n-1) i	f h>1
3 4 5 16 -	
int fib (int n)	fib(5)? = 5
The second secon	all baseness and and siceral
if (n <=1)	Table is generated. Using
return n;	bottom up approach.
F[0]=0; F[1]=1;	- who have been proportionard
for (int i= 2; i = n; i+t)	
{	10/07 NAPELS
F[i] = F[i-2] + F[i-1];	0 + 1
3	T(N) = 2 r (N-1) +1
return F[N];	0/2"31
3	
Danie Bartles ralle ex 1	Company of the Compan
CREVIEW Slide Notes to	Dynamic Programming!)
	(an he used, but most