Alunos: Unicios de Oliveixa Guimerais Vitor Enzo. RA: 802123 DA programação dimâmica consiste m Dolução de Auberdolimo Catada dis	RA: 802431	
DA Programação dimanica comist.	×	1. 1.
Dolução de substicolomo Contratorato disco	a suravicao de p	broblemes a porter da
Delução de subproblems. Entretento, diferentos and or subproblems ao distributos	mummy da Jobra)	tigia di dividir para com:
quistar, and os subproblems são disfundamentos vão compartitar es problemos	and ideas	o dinamista on problima
Dema forma, a idin i ayı or suhanı	oblima alam His	Muida Hamaa uma vaica
Lina forma, a idia i que os subpri rez, tindo sus risultados armazimados dia	tro di um tabila	Data was of something
evitando sutrabalho.	10.00 ON: 0 10 0000.	have bount the enveronment
Assim existem alguns paras para d	homydlur um algarit	imo mando programacão di-
- manua:		· •
1- Definir a astrutura de uma solución	o étima ou in	a the state of the state
2- bitabilian, returni vamente, o val	or de uma salução	otima
3-lalcular o valor de suma soluc	ao otima	
4- lonotruir uma solução otima	a partir da inform	nção calalado.
	1	
		<u></u>
2) Fibonacci (n) { > Calcular	ido a complixidadi	obstruando a precentinicia
$if \eta == 0$ example de	a arvore de recovê	inca:
Juturn o	fib	(7)
if m = = 1	fib(6)	fib (5)
1 Julium 1	fib(5) fib(4)	(ib (3)
else	7.0(3)	10(1)
rulum fibonaco (n-1)+fibonaco (n-2)	fib(4) fib(3) fib(3) fi	b(2) fib(3) fib(2) fib(2) fib(1)
- Je		
12 Observando a arrore de recorrência,	podimos socruju a j	u carrûncia apim:
T(n) = T(n-1) + T(n-2) + O(1)		
Guardo M for para inflimito, T(n	-1) = T(n-2). Qu	aya:
T(n) = 2T(n-1) + o(1)		
$= 2. \left[2. \left[(n-2) + O(1) \right] + O(1) \right]$		
= 2. [2.[2T(n-3)+o(1)]+o(1)	$[1] + o(1) = 2^3 T$	(n-3)+o(1)
	The second secon	The second control of

lontinuando atí K:	
$T(n) = 2^{k} T(n-k) + O(1)$	
	kurpão í T(0), então n-k=0-6 n=k
Logo, times:	
$\pm (m) = 2^{m} \pm (0) + 0(1)$	
O(1)	<u> </u>
Como 2º domino, untão:	
$T(n) = O(2^n)$, sindo assim wim	complitidade exponencial, que i satremamen-
te ruim (PROIBITIVO). Olflando ma árua	u de recursão podemos observas que muitos sil-
- culos satão sendo rualizados diversos o	uzus, sendo assim, provado pelos cálculos, Ine-
-ticinte.	Table 1
3	
a) Abordagem Top-Down (Muma	rização)
Algoritmo da introtigia top-Daum:	Observando o codyo, podemos observar
m[0n] = 0	que o array de volores Colculado i ini-
$\pm ib - M(n)$ {	-ciado com o valor o pora tadas en posi-
if(m==0)(m==1)	- pas e que o memo i prendido confer- -me o cabalos são faitos, dvitando
riturn n	-me os cabalos são fuitos, uvitando
$\inf(M[n] == 0)$	retrabalto.
M[n]:Fib_M(n-1) + Fib_M(n-2) Assim, a complixidade i diminuida consi-
ruturn M[n]	-diraculmente por conta do armazinamen.
three tables of the contraction to the contraction	-to do rusultado dos subproblemas

Data	
b) Abordagim bottom - up (rus)	
Algoritmo da estratigia bottom-up	Obsurvando a função fib-B, Obsura-se
$-$ Eib-B(n) {	que a eficiência de algoritmo i O (n), sendo
if (w==0)	muido melhos do que a farma com eficiêncio
raturn 0;	expanencial.
elsi	
privious = 0	A eficiência dese algoritmo ficer muito
1 toward = 1: - million =	melhor pais no forma iterativa, i possibil
Par (i=1 to n) {	Paladar a siguincia di fibonacci modificando
nui = privious + current	os belores de previous e current conforme os
1 privious = current	calculos são rualizados, bastando um fon de
current = new	j-1 ati n.
	AND THE CONTRACT OF THE REAL PROPERTY.
ruturn current	
about as size of an indeed a	and the second s
4) bra o problema da confegum d	i cidules, podemos viar o siguinte códizo para
Moder o problema:	
F(n)	Observando a arvare de recarrincia:
Laif n==0 (1) a dado	F(7)
ruturn O	F(6)
1 elseximal man miles	
if m==1	F(3) $f(4)$ $F(4)$ $F(5)$
vitum c[1]	
1 1 else	F(1) F(2) F(2) F(3) F(3) F(3) F(4)
sutum max(c[n]+F(n-2), F(n-	3) Nota-re que há muita superição de
1 3	Calcula, ando ineficiente. Vermon Calcular a
	complisidade dese algorismo:
	dourbourner and allo anis.
The second secon	The state of the s

and the second s	
Dodo a siguincia de vicurios sendo $T(n)=T(n)$ Quando m vai para o vinfinito $T(n-1) \approx T(n-2)$	n-1) +T(n-2) + O(1) , center.
Quando myai para o vinfinito T(n-1) = T(n-2)	
logo, T(n)=2T(n-1)+0(1)	
= 2[2T(n-2)+O(1)]+O(1)	
$= 2 \cdot \left[2 \cdot \left[2 \cdot T(m-3) + O(1) \right] + O(1) \right] + O(1)$	
$= 2^3 \cdot T(m-3) + O(1)$	
Continuando K:	
$T(n) = 2^{k}, T(n-k) + O(1), lomo a$	candição di parada i ati T(0):
$T(n) = 2^{k} \cdot T(0) + O(1)$	
$\widetilde{(1)}_{0}$	
T(n)= 0 (zK), que à exponencial (E l'oritanto, como varios cálculos precisam sur referto - vere de recursão), concluirse que O(zK) i	Extrumamente ruin)
Protection como varios cálculos precisam sur ruleito	os diluras laus (Observendo ór-
- I (Les de recursão) condiverse que O(ZK) à	mwia ruim
	Follow Laws to the
3) Aboutagen hotelown - 1112	
Abordagim bottom-up Idimos Jacuus o algaritmo para Doluciemas	o problema do aquíncia de tida
as como sendo:	
n wind hundo.	Amalinando o cooligo, percebenos
ledulas (n){	qui o mismo fassi conflix
Definir um array Fode tamanho M+1	-dade O(n), um rez que
F[a] - 0	ha um laço que percorre
F[1] = C[1] 1239	M-1 reges pendo da
The state of the s	orden o(n).
for i=2 to m F[i]= mmax (c[i]+ F[i-2], F[i-1	1]) Assim, como a complexida.
	HAVII)
EMT	-di (TO(n), usu algoritmo di
raturn F[m] muit	-de at O(n), usu algoritmo de
3 muite	-divi O(n), usu algoritmo ci o mais uficiente do que o nuncial.

6 C=[2/3	5,5,2,10,50,100,50,20,20,50,100]
Inicialmente	timo que F[0]=0 u F[1]=ci=2, Assim timo:
Valor de i	Valor para F[i]
1=2	F[2]= max (C[2]+ F[2-2], F[2-1])= max (5+0, 2)=5
J=3	$F[3] = \max(C[3] + F[3-2], F[3-1]) = \max(5+2, 5) = 1$
j=4	F[4] = max(c[4]+F[4-2], F[4-1]) = max(2+5,7)=7
i=5	F[5] = max (C[5] + F[5-2], F[5-1]) = max (10+7,7)=17
_ j=6	$F[6]= \max(C[6]+F[6-2], F[6-1]) = \max(50+7, 17)=57$
u'=7	F[7]= max (C[7]+ F[7-2], F[7-1])= max (100+17,57)=117
_ v=8	F[9] = max(C[8] + F[8-2], F[8-1]) = max(50 + 57, 117) = 117
i=9	$F[9] = \max(C[9] + F[9-2], F[9-1]) = \max(10 + 117, 117) = 137$
J=10	F[10] = max (C[10] +F[10-2], F[10-1]) = max(20+117, 137) = 137
j=11	F[11] = max (C[11]+F[11-2], F(11-1]) = max (50+137, 137)=187
<u>j=12</u>	F[12] = max (C[12] + F[12-2], F[12-1]) = max (100 + 137, 147) = 237
An annual and an annual and an annual and an annual and an an an annual and an an annual and an an an an an an	
Assim a	valor maximo de dimbero que pode ser colitado sem que 2 cidados
Viginhos exfa	m calitadas i 237.

Digitalizado com CamScanner

luestão_	(FC				-									
Robo-	-ccle	tor (115											
- 10	+ F	0n	1,0	m]	be an_	verrs								
-	F117 =													
t	or)=	2 4	m_o											
	E	Li, L	= FEI,	j-1] + (
+ + (· - i =	2 +	o n	4	- 1				-					
:	FI	- [ارن	FL:-1	,1]+C	(1,1)									
1 1	fo	_ <u>;=</u>	2 +0	_m_c										
		=[- : [;.] :	max	(FC1,], F.	·i,j-12)	+ c[ـــــــــــــــــــــــــــــــــــــــ						
			_											
ret	urn_F	Ln,	ml				-	-		_				
					omplexida Portanto									
mente. (m) e Percebe:	Como	e C O	(n-n) (n-m)	m). > 0 que		a c itmo 3	comp	exid nple: Lup	ade xidad e'	do e e mo:-	al quiv to	gor:tm ale à	0 e ((m).
mente. (m) e Percebe:	Como como se, po idade	e O ortan	(n-n)	m). > 0 que	Portanto (m) entá c algor	a c itmo 3	comp	exid nple: Lup	ade xidad e'	do e e mo:-	al quiv to	gor:tm ale à	0 e ((m).
mente. (m) e Percebe: complex	como se, po idade	e C Ontan O(2	(nm)	m). > O que da	Portanto (m) entá c algor	a c itmo 3	comp com bottom	exid nple: Lup	ade xidad e'	do e e mo:-	al quiv to	gor:tm ale à	0 e ((m).
mente. (m) e Percebe: complex	como se, po idade	e C Ontan O(2	(nm)	m). > O que da	Portanto (m) entá c algor	a c āo a itmo 3	comp com bottom	exid nple: Lup	ade xidad e'	do e e mo:-	al quiv to	gor:tm ale à	0 e ((m).
mente. (m) e Percebe: complex stao O F[nxm	como	e C Ontan O(2	(nm)	m). > Or que da	Portanto (m) entá c algor	a c āo a itmo 3	comp com bottom	exid nple:	ade xidad e'	do e_e	al quiv to	gor:tm ale à	0 e ((m).
mente. (m) e Percebe: complex stão 0 F[nxm	como	e Co	(n·m)	m).	Portanto (m) entá c algor	a canada a decurs	composition	exid nple:	idad	do e_e	al quiv to	gor:tm ale à	0 e ((m).
mente. (m) e Percebe: complex stao 0 F[nxm	como	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	(n·m) (n·m)	m).	Portanto (m) entá c algor	a canada a decurs	composition	exid nple:	idad	do mo:	al quiv to	gor:tm ale à	0 e ((m).
mente. (m) e Percebe: complex F[nxm	como	0 00 00 00 00 00 00 00 00 00 00 00 00 0	(n·m) (n·m)	2 3 4	Portanto (m) entá c algor	a canada a decurs	composition	exid nples	idad	do mo:	al quiv	gor:tm ale à	0 e ((m).

else 1 q = -inffor i = 0 to n $q = max(q, pli) + Cot_Rad_Mem_Aux(p_in-H_r)$ return q

A função Cut-Rod Mem Aux resolve o problems para os tamanhos de 0,1, n. Porem, cado problems é iterado em um for de tamanhon.

O(n²) e sua complex dade

Esse algoritmo em PD e muilo mais eficiente que sua forma recursiva, a qual posso complexidade 0(2")

Questão 10) Resolva para n=6.	
l; 1 2 3 4 5 6	
P; 2 6 9 10 12 16	, e see see see d
	11000
- [0:2,6,9,12,15,18]	9=
$j=1$, $i=1$ $q = max 1 - \infty$, $21 = (2)$	
)=2, i=1 q = max1-002+24=4	
i=2 q= m2x14,6+04=6	
1-3, i=1 q = max1-00,2+61=8	
-3 q=m2x18,9+04=9	
== 1 =1 q=max1-00,2+91=11	
i=2 q-m3x111,6+61-12	
1=3 q-max 112, 9+21-12	
i=4 q= max (12,10+01=(12)	
=====================================	
q= max 4/4,6+91=15	
i=3 q= max 115,9+64=15	
i = 4 q = max 15, 10+24 = 15	
1 = 5 q= max (15,12+04:(15)	
j=6, =1 q= max 1-0,2+154=17	
i=2 q= max 117,6+127=18	
i=3 g= max 118, 9+94=18	
1. + G= max 118,10+61=1B	
1=5 q= max 18, 12+24-18	
i=6 q=max 18,16+01=18	