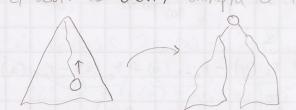
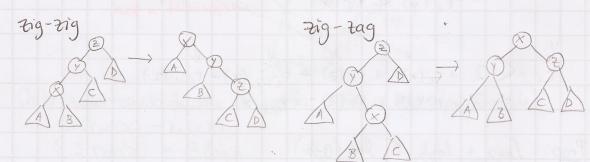
SPLAY TRES

- · Son árboles que se balancian de manera annortizada.
- · Implica que en cada consulta el árbol cambia, en particular, ol consultar un elemento, se harán balcuceos de tal forma que el elto que en la raíz.
- "Una seaevoia de n operaciones tiene costo amortizado de O(log n) amortizado, si todos los eltos tienen prob iniforme /n. El costo es OCH) entropía de Hvfman.

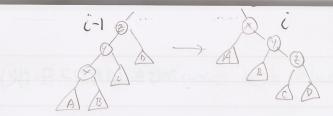


Operaciones de notación (recordar AVL)



- · Zag-tig
- · 20g-20g
- · tig à a dist 1 de · tag) la raíz

Veamos costo de zig-zig...



Fuimos a buscar x

S(x) = tamaño (# de nodos) del subárbol con raíz x (contandox) ri(x) = log_2 S(x) luego de la operación i φi = ξ (π) En un zig-zig la ri() de los nodos on A.B.C.Y.b., and Ci = 2 (dos rotaciones) $\hat{C}i = Ci + \phi i - \phi_{i-1}$ $\hat{c}_{i} = 2 + r_{i}(x) - r_{i-1}(x) + r_{i}(y) - r_{i-1}(y) + r_{i}(x) - r_{i-1}(x)$ Notar que: · (2) = 1: (2) (tienen el mismo &()) · - (4.) > (i-1(x) $\hat{c}_{i} < 2 + r_{i}(x) + r_{i}(z) - r_{i-1}(x) - r_{i-1}(x)$ = 2 + ri(x) + ri(t) - 2k-i(x)Prop: loga + logh & log (a+b) logab & 2 log(a+b) logab < 2 leg(a+6) - 2 log 4 ab & log (a+b) 4 ab & a2 + 2ab + b2 Q a+ 2ab + b2 $0 \le (a-b)^2$

LANIVERSOS DISCHETOS

Lungo $Y_{i-1}(x) + Y_{i}(x) = \log S_{i-1}(x) + \log S_{i}(x)$ $\leq 2 \log S_{i-1}(x) + S_{i}(x) (por prop)$
Como $S_{i-1}(x) + S_{i}(z) \leq S_{i}(x)$
$= \frac{1}{r_{i-1}(x) + r_{i}(x)} < 2 \log \frac{s_{i}(x)}{2} = 2r_{i}(x) - 2$
$r_{i}(x) < 2r_{i}(x) - 2 - r_{i-1}(x)$
$\hat{c}_{i}(x) + r_{i}(x) + 2r_{i}(x) = 2 - r_{i-1}(x) - 2r_{i-1}(x)$
Ci $\langle 3(r_c(x) - r_{c-1}(x)) \rangle$ Nucle a telescópica $3r_{c+2} \cdot 3r_{c+1} \cdot 3r_{c+2} \cdot 3r_{c+1} \cdot 3r_{c+2} \cdot 3r_{c+1} \cdot 3r_{c+1} \cdot 3r_{c+2} \cdot 3r_{c+1} \cdot 3r_{c+2} \cdot 3r_{c+1} \cdot 3r_{c+2} \cdot $
Costo amortizado de una búsqueda Suma (telescópicamente) $3(r_m(x) - r_o(x))$ $53r_m(x) = 3logn$
Analigar para zig-zag, zag-zag, etc es amá ogo en zig y zag da como 3 () + 1 o algo así pero solo 1 vez pres está lobajo do lo raía.

Optimalidad estática.

Si el elemento x se busca g(x) veæs (de las m) entonces el Costo amortitado es $O\left(\sum_{x\in T} \frac{g(x)}{m} \log \frac{m}{g(x)}\right)$

Le caremos un peso w(x) = g(x) = x.

 $w = \sum_{x \in T} w(x) = \sum_{x \in T} \frac{q(x)}{m} = 1$

 $S(x) = \sum_{V \text{ desc de } x} w(x)$, $r_{U}(x) = \log_{Z} S_{U}(x)$

 $\hat{C}_{i} \leq 3 \left(r_{m}(x) - r_{o}(x)\right) + 1$ $= 3 \left(log(\omega) - log(\omega)\right) + 1$

 $= 3 \log \frac{m}{4(x)} + 1$

Si promediamos sobre todos los xo, con su probabilidad
3 9(xx) log xx + 1

1 + 3H.

Universos Discretos

-	Ordenar en O(n) (voivers
-	Predecesor en O (loglojo) Tries y árboles de sufijos natings
-	Tries y arboles de sufijos natings +2 +2 +2 +2
	9 (m) 1 (m)
	2 million by purious of a standards at a contract of the contract of
@	Counting Sort
	- las claves estain en [1U]
	-n claves no how registros asociados
	- no hay hada más que las clares (no hay punteros. Dos numeros con el mismo valor son indistinguible → inicializar contadores C[i] ← 0
	t a cumular el # de ocurrencias de cada clave
	-> output el # de veces que aparece cada clave.
	The second of the second control of the seco
	for i < 1 to U
	CE: 2 = 0 A= 32115213122
	for ja 1 to n 11112222335
	C[A[i]] + C[A[i]]+1 C: 1 D
	for it 1 to U is plate of (all soon 3) I be done of anxion
	for k = 1 to C[i].
	Bendin AEj 76-0 admit (1/2 0) 0 / de la constant
	Establish $j \in j+1$ and a subject $(V+D) \in j+1$
	Funciona porque los números iguales son indistinguibles. Los elementos SON la clares Complejidad O(n+U) Conviene solo ciando U es razonable. Gasto mucho espacio
	extra.

Uso (los elementos sen algo más que solo clares) Bucket Sort 1º ciento 2º inicializa proteros a la zona de A donde se escribentos distritos cloros 3º paso por A copiando cada voltor a su posición definitura en A A = 32 [1] 5 2 13 1 2 2 Conserva la identidad de los eltos y el alg. es estable (preserva el orden relativo complejitad P(n + V) también Complejitad P(n + V) también L' inicializar contadores L'ego podemos erdurar en tiempo lineal si [V] es P(n) Bajo un modelo NO pasodo en comparaciones C' falemos ir más lejas?		214 202 144	
constantes clores distintos clores 3 paso por A copiando cada voltor a su posición difinitud en A A = 3 2 11 8 2 13 1 2 2 foi (1-1+0 U) CI 1 pos 8 1 2 1	and les elements si son distinguibles us (los elementos son algo más que se Bucket Sort	(vando tienen = valor do clares)	
constantes clores distintos clores 3 paso por A copiando cada voltor a su posición difinitud en A A = 3 2 11 8 2 13 1 2 2 foi (1-1+0 U) CI 1 pos 8 1 2 1	1º crento		
3° paso por A copiando cada valor a su posición definitud en A' A = 32/152/3122 for i = 1+0 U Clize O Clize O For j = 1+0 n Clarcol + Clarcol + 1 Pos 11 Pos 11 Pos 11 Pos 11 For i = 2 to U Plize - Plize - Pl+ Ci-17 A' = 1/2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 inicializo protenos a la Zona de A	conde se esch ben los	
A = 32 [15 2] 3 22 for it 1to U Clist O For j = 1 to n CLARICO] + CLARICO] + 1 Pros M Pros M Pros M Pros M Pros M For i = 2 to U Pri I = Pri - 11 + Cri - 17 For i = 2 to U Pri I = Pri - 11 + Cri - 17 A = 1/2 1	distintos clones	tox ant po	
A = 32 [15 2] 3 22 for it 1to U Clist O For j = 1 to n CLARICO] + CLARICO] + 1 Pros M Pros M Pros M Pros M Pros M For i = 2 to U Pri I = Pri - 11 + Cri - 17 For i = 2 to U Pri I = Pri - 11 + Cri - 17 A = 1/2 1	3 paso por A copiando cada vollor c	Su posicion definitiva en A	
C: 1 [] postility of C[i] to n 3 L postility of C[Ario] to n C[Ario] C[Ario] to n C[Ario] C[Ario] to n C[Ario] C[Ario] to n P[i] to n P[i			
Conserva la identidad de los eltos y el alg. es estable (preserva el orden relativo Complejibad O(n + 0) tambén Liago Podemos ordinar en tiempo lineal si V es O(n) Bajo in modelo NO basodo en comparaciones (Podemos ir mas lejos?			
Conserva la identidad de los eltos y el alg. es estable (preserva el orden relativo Complejibad O(n + 0) tambén Li inicidizar contodores Luego Podemos er durar en tiempo lineal si [U] es O(n) Bayo m modelo NO pasado en comparaciones C'Podemos ir más lejos?	C: 1 D pas 4 5	(1.74 O	
Conserva la identidad de los eltos y el alg. es estable (preserva el orden relativo Complejibad O(n + 0) tambén Li inicidizar contodores Luego Podemos er durar en tiempo lineal si [U] es O(n) Bayo m modelo NO pasado en comparaciones C'Podemos ir más lejos?	2 17 25 86789	for i 1 to n	
PCIJE 1 S 1 pos 11 S 1 pos 11 A 2 12 12 12 12 12 13 13 13 13 for je 1 to n A 2 17 1	3 L tos 9 10 11	05. 74050 -7.	
A'= 14 1 1 1 1 1 2 2 2 2 3 3 3 15 for je 1 to n A' EP [A []] I A E] P[A []] A EP [A []] I A E] Conserva la identidad de los eltos y el alg. es estable (preserva el orden relativo original entre la inicidirar contadores Complejibad O(n + 0) también Li inicidirar contadores Lugo Podemos er duras en tiempo lineal si [U es O(n)] Bajo m modelo NO pasado en comparaciones C' Podemos ir mas lejos?	4 20511	Pr.7 ← 1	
A'= 14 1 1 1 1 1 2 2 2 2 3 3 3 15 for je 1 to n A' EP [A []] I A E] P[A []] A EP [A []] I A E] Conserva la identidad de los eltos y el alg. es estable (preserva el orden relativo original entre la inicidirar contadores Complejibad O(n + 0) también Li inicidirar contadores Lugo Podemos er duras en tiempo lineal si [U es O(n)] Bajo m modelo NO pasado en comparaciones C' Podemos ir mas lejos?	5 1 pos 11 2	for it 2 to U	
Conserva la identidad de los eltos y el alg. es estable (preserva el orden relativo Complejibad O(n + 0) también original entre daves ignales) Liego Podemos er durar en tiempo lineal si (V) es O(n) Bajo n modelo NO pasado en comparaciones C'Podemos ir mas lejos?	La		
Conserva la identidad de los eltos y el alg. es estable (preserva el orden relativo Complejibad O(n + 0) también original entre daves ignales) Liego Podemos er durar en tiempo lineal si (V) es O(n) Bajo n modelo NO pasado en comparaciones C'Podemos ir mas lejos?	A= 41 1 1 1 2 2 2 2 3 3 5	for je 1 to n	E
Conserva la identidad de los eltos y el alg. es estable (preserva el orden relativo Complejibad O(n + 0) también original entre daves ignales) Liego Podemos er durar en tiempo lineal si (V) es O(n) Bajo n modelo NO pasado en comparaciones C'Podemos ir mas lejos?	1 2 3 4 5 6 7 8 9 10 11	A'EPEACITITE AGI	E
Complejibad O(n + 0) también original entre daves iguales) Liego Podemos ordurar en tiempo lineal si Mes O(n) Bajo n modelo NO pasado en comparaciones C Podemos ir más lejos?		PEAGJ] - PEAGJ]+1	E
Complejibad $O(n + O)$ también original entre daves ignales) = Liego Podemos ordurar en tiempo lineal si V es $O(n)$ Bajo n modelo NO basado en comparaciones C Podemos ir más lejos?	Conserva la identidad de los eltos y el		
Liego Podemos er devar en tiempo lineal si Wes O(n) Bajo n modelo NO pasado en comparaciones C Podemos ir mas lejos?		orden relativo	
Liego Podemos orderar en tiempo lineal si Wes O(n) Bajo n modelo NO basado en comparaciones C Podemos ir mas lejos?	Complexited O(1 + 0) toursien	91,41,109	
DODDE COUNT MEDICA DE COMBETA DE	Lini adlizar contador	es dave guales).	-
DODDE COUNT MEDICA DE COMBETA DE		si Mar Ala	E
DODDE COUNT MEDICA DE COMBETA DE	Liego podemos eravar en trempo lineal	SI (UTE) UCII)	F
DODDE COUNT MEDICA DE COMBETA DE	dela alla pare loine?	a cours of the	E
	Closemos Ir 1140s (egos).		E
		- 101	

17 10001 Varmos a ordenar for el vitimo dit viando 5 00101 bucket sert (1V1=21) 2 00010 14 01110 Athora por el siguido 6 00110 2 00010 8 01000 ahora estan 8 01000 17 10001 5 00101 positios. 7 5 00101 5 00101 positios. 7 5 00101 5 00101 positios. 7 8 01000 -> 17 10001 5 00101 positios. 7 8 01000 -> 17 10001 5 00101 positios. 7 8 01000 -> 17 10001 5 0010 ALMISMO 17 10001 -> 17 10001 2 00010 2 00010 Pres bucket 5 00101 5 00101 Sert es ESTABLE 6 00110 8 01000 14 01100 8 01000 14 01110 9 00010 15 00010 19 01110 19 01110 19 01110 19 01110 19 01110 19 01110 19 01110 19 01110 10 001				
5 0 0 1 0 1 bucket sert (1 1 1 = 2 1) 2 0 0 0 1 0	17 1000	Namos a octomos	mol sitimo	L'Alla Call
200010 14 01110 Athora por el segundo 8 01000 6 00110 8 01000 ahora estan 8 01000 17 10001 por los 2 5 00101 14 01110 Cltmos 2 00010 bits 2 00010 TEMPO 17 10001 5 0010 TEMPO 17 10001 5 00101 Sert es 14 01110 6 0010 ESTABLE 6 00110 12 01000 TEMPO 14 01110 6 00110 ESTABLE 6 00110 12 0000 TEMPO 14 01110 GOOD TEMPO 14 01110 TO TEMPO 15 00101 TO TEMPO 16 00110 TO TEMPO 17 01001 TO TEMPO 18 01000 TO TEMPO 19 01010 TO TEMPO 19 01010 TO TEMPO 10 0010 TO T	500101	bucket sort (11	11=21)	or wands
200010 20010 Ahora por el sigundo 801000 17 10001 20010 Ahora por el sigundo 801000 17 10001 por lles 7 50001 1401110 por lles 7 50001 1401110 ALMISMO 1710001 -> 1710001 por lles 7 1401110 60010 Pres bucket 500101 50010 ESTABLE 600110 20010 ESTABLE 600110 1401110 Cuántos bits escribir 900010 140110 Cuántos bits escribir 900010 190110 Pres por elegir chunks bits escribir 100010 190110 Pres por elegir chunks has greunos que 1. Redo elegir chunks de login bits para que el miverso sea n (fecordar que Buchet sort Runciana en O(n+U) si Ul er O(n)	14 01110			
8 01000 6 00110 8 01000 ahora estans 8 01000 17 10001 por los 7 5 00101 19 01110 contenados 17 10001 5 00101 por los 7 5 00101 19 01110 contenados 17 10001 19 0110 bits 18 01000 17 10001 por los 7 18 01000 17 10001 per bucket 5 00101 5 00101 sert es 14 01110 6 00110 ESTABLE 6 00110 8 01000 per bucket 5 00101 8 01000 per bucket 5 00101 8 01000 per bucket 6 00110 19 01100 per bucket 19 01100 per per no region degir churks más greundes que 1? Redo degir churks de logh bits para que el miverso sea n (Accordar que Bucket sort funciana en O(n+V) si V er O(n)	200010	14011110	Ahora por el se	avnda
8 01000 17 10001 ordenados 17 10001 5 00101 por los 7 5 00101 14 01110 c/thmos 2 0 0010 bits b 0 0110 ALMISMO 17 10001 -> 17 10001 pres buckt 5 00101 5 00101 sert es 14 01110 6 00110 ESTABLE 6 00110 8 01000 10 contos bits escribir 9 2 00010 8 01000 10 contos bits escribir 9 2 00010 10 0 (n log V) los dementos 19 0110 10 10 10 10 10 10 10 10 10 10 10 1		2000/10		
17 10001 5 00101 por los 7 5 00 01 14 0 11 10 0 01 15 00 10 2 0 00 10 0 0 15 0 0 10 0 0 0 0 0 0 0 0 0	801000			ahora estan
500 (01 14 011 10 VITIMOS 2000 10 bits 1710001 -> 17 10001 2000 10 THEM PO 2000 10 Pres bucket 500 101 Sert es 1401110 600110 ESTABLE 600110 20010 2000 Hamber Service 1401110 600110 ESTABLE 600110 20010 2000 Hamber Service 1401110 110 Hamber Service 1401110 Hamber Service 1401110 Hamber Service 1401110 Hamber Service 10001 Hamber Servic				ordenados
2 0 0 0 10 bits 8 0 10 0 0 17 0 0 0 17 0 0 0 17 0 0 0 17 0 0 0 17 0 0 0 17 0 0 0 17 0 0 0 0				
17 10001 -> 17 10001 2 00010 2 00010 3 0100 4 Mismo Tiem RO Pies bucket Social Social Social Social 6 00110 8 01000 14 01110 6 00110 9 countes bits escribir				
17 10001 -> 17 10001 2 00010 2 00010 3 00101 5 00101 6 00110 6 00110 7 EM PO Res bucket Sert es ESTABLE 6 00110 9 (n log V) 10 outlos 10 out	\rightarrow 00000			
2 0 0 0 10 5 0 0 10 1 5 0 0 10 1 5 0 0 10 1 Sert es ESTABLE 6 0 0 1 1 0 14 0 1 1 1 0 6 0 0 1 1 0 7 countes bits escribir 8 0 1 0 0 0 8 0 1 0 0 0 14 0 1 1 1 0 14 0 1 1 1 0 14 0 1 1 1 0 15 10 0 0 1 16 0 0 1 1 0 17 0 1 1 1 0 18 1 0 0 0 1 19 0 1 1 10 19 1 0 0 0 1 Predo elegar chunks de logh bits para que el universo sea h (Fecordar que Bucket sort funciona en O(n+V) si V er O(n) 10 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			000110	
Sobject 5 00101 Sort es 140110 6 00110 ESTABLE 600110 May 01110 200010 V On lay 0) Counter bits escribir 801000 140110 V On lay 0) Counter bits escribir 801000 140110 V On lay 0) Counter más grandes que 1? Redo elegar chunks de layabits para que el miverso sea h (Pluordar que Bucket sort funciona en O(n+U) si U er O(n)				
14 0 11 10 6 0 0 110 ESTABLE 6 0 0 1 10 1 10 10 10 10 10 10 10 10 10 1				sort es
Predo elegar chanks de logh bits para que el miverso sea h (fluordar que Bucket sort funciona en O(n+V) si V er O(n)		6 00110		
Redo clogir chinks de loya bits para que el miverso sea n (fecordar que Bucket sort funciona en $\Theta(n+U)$ si $ U $ er $\Theta(n)$	600110	8 01000		
14 0 1 1 10 0 0 1 1 1 1 1 0 0 0 1 1 1 1		14 0 1110		
14 0 1 1 10 0 0 1 1 1 1 1 0 0 0 1 1 1 1	All Mark and All M	and the selection of	- cuantas bits	escribit
14 0 1 1 10 0 0 1 1 1 1 1 0 0 0 1 1 1 1	500101	96.0	& outener	tos
14 0 1 1 10 10 10 10 11 10 11 10 10 10 10		· (1) log (05	
14 0 1 10 0 0 11 La por que no rrejor elegir chunks más greunder que 1? Puedo elegir chunks de loya bits para que el miverso Sea n (fecordar que Bucket sort funciona en O(n+U) si U er O(n) 2) O(1 11) O(1 11)				
Predo elegir chanks de loya bits para que el miverso sea n (fecordar que Bucket sort funciona en O(n+U) si U er O(n)				
(fecordar que Bucketsort Runciona en O(n+U) si U er O(n)	19 10001			
(fecordar que Bucketsort Runciona en O(n+U) si U er O(n)	To por g	é no mejor elegir	chunks más gre	moes que 1?
De de Broketsort Rinciona en O(n+V) si V er O(n)	ruedo elegir chunks d	e loya bits para qu	e el miverso Sa	ra h
	(Hoordar que Buch	etsort funciona	en O(n+U')	Si U er Da)
050 wands Ue lay grande proble legar a competir con affect sort.	=) P(n loa)	1) - 0-1		
050 cuando U es May grande puede legar a competir con aviet sort.	On the	e 1 - Unlight) -> 0	(n log n 6) = 0 (6 h)
	050 ciando O e mais	y grande	prede llegar a compet	is con Quicksort.