עבלת גיבוב Hash Table
• מה נלמד? • מבוא • משלת בהן תומכת טבלת גיבוב • דוגמאות לשימוש בטבלת גיבוב • גיבוב עם שרשור • גיבוב עם שרשור • ביבוב עם שרשור • בחירת פונקציית גיבוב טובה
מבוא - באפליקציות רבות עלינו לתחזק קבוצה <i>דינאמית S</i> לאחסון איברים, כך שנוכל לאתר אותם בעזרת מפתח <i>ייחודי</i> שמצורף לכל איבר בעזרת מפתח <i>ייחודי</i> שמצורף לכל איבר
$oldsymbol{\chi}$ $oldsymbol{key}$ $oldsymbol{Satellite}$ data קבוצת עובדים במכללה קבוצת המשתנים בקוד קבוצת אתרים זדוניים
המפתח - ת"ו של עובד המפתח - שם המשתנה המפתחות - כתובת האתר
IP Addresses

	 מילון		
	תומך בשלוש פעולות עיקריות Dictionary		
	בהינתן מפתח k , הפעולה מחזירה איבר - Search(k) •	x.key = k מקיים	
	אם אין איבר כזה NULL או		
	x הכנסת לקבוצה את איבר – $Insert(x)$		
	בהינתן איבר x , יש למחוק אותו מהקבוצה –Delete (x)		
4			
	_		
	טבלת גיבוב הבטחה על זמני ריצה		
	א א $Search(k)$ א $O(1)$ $Search(k)$ $Insert(x)$		
	O(1) Insert(x) • Delete(x) •	ומש נכון	
	Detecte(x)	•	
5			
	_		
	טבלת גיבוב טבלת גיבוב		
	דוגמאות לשימוש		
	D. D.		
	var (103 212 162 56)		
	בהינתן כתובת IP בהינתן שם ב לבדוק האם המשתנה, ה	בהינתן מספר הטלפון, לשלוף	
	האתר נמצא לאתר את הערך י	את הכתובת אה פיצה)	

	<u> </u>
	 טבלת גיבוב
	דוגמא לשימוש
	S_2 (מון: שתי קבוצות של מספרים שלמים S_2 -1. S_2 בגודל S_2 המטרה: לבדוק האם $S_2=S_1$
	פתרונות
	$O(n)$. נאיבי: 3 $O(n^2)$. נאיבי:
	• עבור כל איבר מ- S_2 בדוק האם הוא • הכנס את איברי הקבוצה S_1 לטבלת נמצא ב- S_2 גיבוב S_2
	• עבור כל איבר מ- S_2 בדוק האם הוא • $O(n \log n)$
	- פייין את S_1 נמצא ב-H י עבור כל איבר נר S_2 בדוק האם הוא
	(חיפוש בינארי) S_1 -נמצא ב
7	•
'	
	- המטרה
	. $Search(k)$ *
	Insert(x)*
	Delete(x) \circ
	$\emph{U} = \{0.1,,u\}$ הנחה: המפתחות נלקחו מתוך קבוצה אוניברסאלית
	•
	•
8	
	_
	טבלת מיעון ישיר
	Direct address table
	- נתחיל מטכניקה פשוטה שתוביל אותנו לרעיון של טבלת גיבוב
	• <mark>טבלת מיעון ישיר</mark> היא טכניקה מאוד פשוטה לייצוג קבוצה דינאמית של איברים
	•
9	
-	

T[0 U -1] לייצוג S נשתמש במערך $T[0 U -1]$ לייצוג S נשתמש במערך S לכל תא ב- S תאים מפתח S לכל תא ב- S תאים מפתח S S תאים S לכל תא ב- S תאים S תאים S
ארנת האים מפתח מ-עת מת אים אינד מתאים אינד מת אות אינד מת אות אות אות אות אות אות אות אות אות או
$T[i] = NULL \text{ norm} \cdot \frac{v}{ v }$ $\frac{v}{ v } = \frac{v}{ v } \text{ satellite data}$ $\frac{v}{ v } = \frac{v}{ v } \text{ satellite data}$ $\frac{v}{ v } = \frac{v}{ v } = v$
V
99 40 70 8 3 4 2 3 3 4 2 5 5 5 6 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
10
5 8 8 8
8 8
טבלת מיעון-ישיר <u></u>
Search(k) satellite data
U(1) return $T[k]$
Insert(x) $\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$
$T[x.key] \in x \qquad \begin{bmatrix} 1^{9^4} & 4^{9^4} & 7^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & 1^{10} & 1^{10} \\ 1^{10} & 1^{10} & $
Delete(x)
$0(1) T[x.key] \leftarrow NULL $
מה יכולה להיות הבעיה?
—— טבלת מיעון-ישיר
סברות נויעון-ישיו הבעיה
P Departs
103.212.162.56
קבוצת עובדים קבוצת אתרים אם U גדולה, T באוניברסיטה T דוניים לאחסן טבלת T
קבוצת עובדים קבוצת אתרים אם U גדולה,

1		
	עבלת מיעון-ישיר	
	הבעיה מטרה: לתחזק קבוצה <i>דינאמית S</i> לאחסון איברים, כך שנוכל לאתר אותם	
	בעזרת מפתח <i>ייחודי</i> שמצורף לכל איבר.	
	- קבוצת עובדים באוניברסיטה - 100 אפשרויות לתעודת זהות (אפשרויות לתעודת זהות (אפשרויות לתעודת זהות (אפשרויות לתעודת זהות (אפשרויות לעודת זהות (אפשרויות (אפשרויות (אפשרויות (אפשרויות (אפשרויות (אפשרויות (אפשרויות (אפשרויות (אפשרויות (או אפילו לא אפשרוי) (אפשרויות (או אפילו לא אפשרויות (אפשרויות (או אפילו לא אפשרויות (או אפילו לא אפשרויות (אפשרויות (אפשר	
13		
	 טבלת גיבוב	
	הרעיון	
	י ליצוג קבוצה S נשתמש במערך $m = \Theta(S)$, $T[0m-1]$ $m = \Theta(S)$, $T[0m-1]$ א לכלי S המאים מספר בין S לכלי S המאים מספר בין S לכלי S (hashing larger at vicinity) S (hashing l	
	מה יכולה להיות הבעיה?	
14	_	
ı		
	סבלת מיעון-ישיר הרעיון ַ	
	י ליצוג קבוצה S נשתמש במערך $m = \Theta(S)$, $T[0m-1]$ $m = \Theta(S)$, $T[0m-1]$ $h(k_1)$ $m-1$ לכל S מיפוי מה נקרא S ניבוג S מיפוי מה נקרא S (hashinction) S (hash function) S (hash S	
	(collisions) מה יכולה להיות הבעיה?	

$h(x) = h(y)$: $x,y \in U$ בנור מפתחות שונים
נכון או לא נכון: טענה: לא ניתן למנוע לחלוטין את ההתנגשויות. 1. הטענה לא נכונה 2. הטענה לא נכונה
17
התנגשויות $h(x) = h(y): x,y \in U$ התנגשות: עבור מפתחות שונים $x,y \in U$ מיעון פתוח (open addressing) פתחנות בת מיעו פתוח (chaining) בי מיעון פתוח (איבר אחד בכל תא או $x,y \in U$ בהמנטה בוחנים בזה אחר זה בהמנטה בוחנים בזה אחר זה האים עד שמוצאים תא ריק היק בי בהמנטה בודירה סדרה של $x,y \in U$ בי בענים בודירה סדרה $x,y \in U$ בי בענים בענים בי בענים בודירה סדרה $x,y \in U$ בי בענים בענים בי בענים בענים בענים בי בענים

	שיטת השרשור	•
	h(x) = h(y) (געור מפתחות שונים: אונים התנגשות: עבור מפתחות	1
	(chaining) שיטת השרשור	•
_	- התא <i>ן</i> מכיל מצביע לראש הרשימה של כל האיברים המאוחסנים המוברבים ל- <i>ן</i> 	
	אם אין איברים כאלו NULL אם אין איברים כאלו או NULL או NULL או או איברים אין איברים אינו אינו אינו אינו אינו אינו אינו אינו	
	K ke	
	k_2 k_3 k_3 k_3 k_3	
	/MIE NA	
		19
	ניינווש וופעת וונ	•
_	Insert (T,x)	
	Insert x at the head of the list $T[h(x.key)]$	
	Search (T, k) Search for an element with key k in list $T[h(k)]$	
	Delete (T, x)	
	Delete x from the list $T[h(x.key)]$	
		20
		20
	_	
	דוגמה	r 1
	$m=5$ יש להכניס את המפתחות 6,9,19,26,30,106,309 לטבלת גיבוב בגודל $h(k)=k \bmod 5$ באמצעות שיטת השרשור עם פונקצית גיבוב	
	0	
	1 6 /	
	3	
_	4 9 /	

	 דוגמה		
	יש להכניס את המפתחות 5,30,106,309 יש להכניס את המפתחות 5,30,106,309	m=5 לטבלת גיבוב בגודל 6,9,19,26,	
	באמצעות שיטת השרשור עם פונקצית	$h(k)=k\ \mathrm{mod}\ 5$ גיבוב	
]	0 30 /	
		1 6 /	
	7	3	
		4 9 /	
22			
	דוגמה		
	יש להכניס את המפתחות 3,30,106,309 באמצעות שיטת השרשור עם פונקצית		
	7 = p. 13		
	6 /	$ \begin{array}{c c} 0 & \longrightarrow & 30 \\ 1 & \longrightarrow & 26 \\ \end{array} $	
		. 2	
	9 /	3 4 19 -	_
23			
	נתונה טבלת ניבוב $ au$ שבה הח	תנגשויות נפתרות על ידי שרשור.	
	נסמן ב- m את גודל הטבלה וו	ב- n את מספר האיברים בטבלה.	
	השלימו את המשפט: זמן ריצה במקרה	ה הגרוע של פעולת <i>Insert</i> הוא:	
		'	
.1	.2	.4	
	$\Theta(m)$ $\Theta(n)$	נמצא ביחס ישיר לאורכה של הרשימה $\Theta(1)$	
		שבתא - י כל הו שינווו שבתא	
24	I	, I I	

ידי שרשור. את בתכלה.	תנגשויות נפתרות על ב- n את מספר האיבר	כלת גיבוב T שבה הו עת נודל הטכלה ו	נתונה טו
	II KII IJOII JIK // -I	in a since a s	
<i>ו</i> הוא:	nsert הגרוע של פעולת ז	משפט: זמן ריצה במקרו	השלימו את הו
.4	.3	.2	.1
· נמצא ביחס ישיר לאורכה של הרשימה	Θ(1)	$\Theta(m)$	$\Theta(n)$
שבתא			
•	1	'	25
	תנגשויות נפתרות על		
ים בטבלה.	ב- n את מספר האיבר	את גודל הטבלה ו m	נסמן ב-
.s הוא:	earch הגרוע של פעולת i	משפט: זמן ריצה במקרו	השלימו את הו
•			
.4	.3	.2	.1
נמצא ביחס ישיר לאורכה יייל ברייימר	Θ(1)	$\Theta(m)$	$\Theta(n)$
של הרשימה שבתא			
•		I	26
			20
ידי שרשור.	תנגשויות נפתרות על	כלת גיבוב <i>T</i> שבה הו	נתונה טו
ים בטבלה.	ב- n את מספר האיבר	את גודל הטבלה ו m	נתונה טו נסמן ב- י
3 הוא:	earch ז הגרוע של פעולת	משפט: זמן ריצה במקרו	השלימו את הו
 . 4	.3	.2	.1
נמצא ביחס ישיר לאורכה			
 של הרשימה שבתא	Θ(1)	$\Theta(m)$	$\Theta(n)$
•			6-
			27

?	Tנתונה סבלת גיבוב נסמן ב m את גודל	וב <i>T</i> שבה התנ דל המכלה וכ-	נגשויות נפתרות ע את מספר האור	' י די שרשור. בום במכלה
	Luank W-Thon	-111171011 /1	rkii iyoli jik // -	.11 202 11.
הי	זשלימו את המשפט: זמן ריי	ון ריצה במקרה ה	lete הגרוע של פעולת	:חוא <i>De</i>
_				
.1	.2	3	.3	.4
ı)	(m) $\Theta(n)$	Θ(m	Θ(1)	נמצא ביחס ישיר לאורכה של הרשימה שבתא
				GEITH
28				
0	נתונה טבלת גיבוב <i>ד</i> נסמן ב- m את גודל ו			
הי	זשלימו את המשפט: זמן ריי	ון ריצה במקרה ה	lete הגרוע של פעולת	ם הוא: De
.1	.2	3	.3	.4 נמצא ביחס
ı)	(m) $\Theta(n)$	Θ(m	Θ(1)	ישיר לאורכה של הרשימה
				שבתא
29				
מינ	- מוש הפעולות			
				Lucand (M.)
	θ(1)	T[h(x.key)]	the head of the list	
igth)	(k)] O(list leng		n element with key i	
igth)	Θ(list leng	v)]	om the list $T[h(x.key)]$	Delete (T,x) Delete x fr
	אם <i>m</i> הוא גודל טבלח			טבלה,
	אורך הרשימה יכול לה	ל להיות כד מטפו	ר בין <u>∺</u> 7- <i>n</i>	
30				

 ניתוח של גיבוב עם שרשור	
- הנחות	
1. פונקציית גיבוב צריכה לקיים את הנחת <mark>הגיבוב האחיד הפשוט:</mark>	
ההסתברות שמפתח כלשהו יגובב לתא מסוים שווה עבור כל m התאים, ואינה תלויה בערכי הגיבוב של האיברים האחרים.	
$\Theta(1)$ 2. הזמן הדרוש לחישוב פונקצית גיבוב הוא	
 פעולות חיפוש, הכנסה, מחיקה כוללות חישוב של פונקציית ניבוב 	
-	
-	31
	-
_	
- נגדיר $\dfrac{\alpha}{m}$ מקדם העומס (load factor) – המספר הממוצע של איברים המאוחסנים	
ברשימה מקושרת אחת	
α · כול להיות קטן מ-1, שווה ל-1 או גדול מ-1 α יכול להיות קטן מ-1, שווה ל-1 או גדול מ-1	
$ \begin{array}{c c} 1 & \longrightarrow & 106 & \longrightarrow & 26 & \longrightarrow & 6 & \nearrow \\ 2 & & & & & & & & & & & \\ \end{array} $	
3 4 309 19 9 $\alpha = \frac{7}{2} = 1.4$	
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
-	32
_	
ניתוח של גיבוב עם שרשור	
- ננתח זמן ריצה של חיפוש כושל - זמן ריצה של חיפוש מוצלח ומחיקה חסום על ידי זמן ריצה של חיפוש כושל	
-	
 _	
-	33

	משפט: בטבלת גיבוב שבה התנגשויות נפתרות על ידי שרשור, זמן ריצה הצפוי	
	נושפט: בטבז תגיבוב שבה הונתגשויות נפותו ות על ידי שו שוו , זתן דיצה הצפוי $\Theta(1+a)$	
	- נניח מחפשים איבר x שלא נמצא בטבלת הגיבוב	
	$\Theta(1)$ - $h(k)$ חישוב פונקציית גיבוב •	
	$\Theta(list\ length)$ - $T[h(k)]$ מעבר על הרשימה שבתא -	
	- תחת ההנחה של גיבוב אחיד ופשוט, תוחלת אורך הרשימה שבתא היא α	
	- מכאן, זמן ריצה של חיפוש כושל הוא $(1+lpha)$	
0.4	-	
34		
	_	
	- ניתוח של גיבוב עם שרשור	
	משפט: בטבלת גיבוב שבה התנגשויות נפתרות על ידי שרשור, זמן ריצה הצפוי	
	$\Theta(1+lpha)$ של חיפוש כושל הוא $ heta$	
	$\Theta(1)$ אם $lpha=\Theta(1)$ אם הריצה של חיפוש כושל הוא א נקבל שזמן הריצה של איפוש כושל אי	
	- תחת איזה תנאי $lpha = \Theta(1)$ יתחת איזה תנאי מידל מודל מודל מודל מודל מודל מודל מודל מו	
	m , m הוא גודל טבלת הגיבוב ו n הוא מספר האיברים בטבלה m , m (n $=$ $\theta(1)$ הוא $\theta(n)$ $=$ $\theta(n)$	
	•	
35	-	
33		
	_	
	מהי פונקציית גיבוב טובה?	
	-	
	. ביצועים טובים • פונקציית גיבוב צריכה לקיים את הנחת <mark>הגיבוב האחיד הפשוט:</mark>	
	ההסתברות שמפתח כלשהו יגובב לתא מסוים שווה עבור כל m התאים.	
1	. חישוב מהיר <mark>(1)⊖</mark> • פעולות חיפוש, הכנסה, מחיקה כוללות חישוב של פונקציית ניבוב	
	-	
	_	
36	-	

פונקציות גיבוב גרועות דוגמא 1 • מפתחות: מספרי טלפון (הזמנת פיצה ביישוב קטן) • מספר נייד מכיל 10 ספרות, $ U = 10^{10}$ • $m = 1000$ • פונקציית גיבוב מאוד גרוע: • $h(k) = first\ 3\ digits\ of\ k$	37
 פונקציות גיבוב גרועות	
IP Addresses	
103.236.162.56	
162.252.172.41	
• מפתחות: כתובות IP (רשימה שחורה של אתרים) ו180.181.68.221	
203.134.40.41 $ U = 2^{32}$ ביטים, 32 ביטים (U	
• פונקציית גיבוב גרוע:	
h(k) = last segment of the address (last 8 bits)	
h(103.256.162.56) = 56	
<i>m</i> = 256 ∗	
 החלק האחרון של כתובת IP בדרך כלל מייצגת מספר קטן (חד או דו-ספרתי) 	
-	
;	38
פונקציות גיבוב גרועות –	
- 3119111 212 1111 21213	
2 111217	
- דוגמא 3 22 . היא קבוצת מספרים שלמים זוגיים $5 \cdot$	
 -	
$m = 1000 \cdot$ $h(k) = k \mod 1000$ פונקציית גיבוב גרוע:	
 - 2 46 $h(k) = k \mod 1000$ פונקציית גיבוב גרוע: $h(k) = k \mod 1000$ - מובטח שתאים בעלי אינדקס אי-זוגי יהיו ריקים a	
•	
30 94	
 -	
 <u>-</u>	
	39
•	J

—— שיטת החילוק (Division method)	
h(k) = k mod m	
ה בודל הטבלה – תוכם או בודל הטבלה – תוכם היו בודל הי	
0 30 S = {6,9,19,26,30,106,309} •	
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	
3	
4 309 19 9	
	40
2	40
— (Division method) שיטת החילוק	
h(k) = k mod m הודל הטבלה $-m$	
· • פשוט • מהיר	
. היזהר" מערכים מסוימים של m לבחור את m להיות ראשוני -	
	4.4
2	41
$h(k) = \lfloor m(kA mod 1) floor$ בוודל הטבלה בוודל הטבלה $-m$ בווע $-m$	
13456.09 mod 1 = 0.09	
$57.891 \mod 1 = 0.891$	
	4.0
	42

	(Multiplicatio	—— שיטת הכפל (method	
	$h(k) = \lfloor m(kA mod - k) - k nod $	1)]	
		דוגמה	
	$h(123456) = [16384(123456 \cdot 0.618 \mod 1)] =$	k = 123456	
	= [16384(76295.808 mod 1)] = = [16384 · 0.808] = = [13238.272] = 13238	$m = 2^{14} = 16384$ $A = (\sqrt{5} - 1)/2 \approx 0.618$	
		43	ł
		40	,
		_	
		התנגשויות	
	$h(x) = h(y)$: $x,y \in U$ פים	התנגשות: עבור מפתחות <u>שו</u>	
		פתרונות	
	2. מיעון פתוח (open addressing) 1- רק איבר אחד בכל תא או <i>NULL</i>	1. שיטת השרשור (chaining)	
	$m{m}$ פונקציית גיבוב מגדירה סדרה של $h(k,0),h(k,1),\dots,h(k,m-1)$ אינדקסים בזה אחר זה תאים עד בהכנסה בוחנים בזה אחר זה תאים עד		
	שמוצאים תא ריק $lpha \le 1$ מקדם העומס $lpha \le 1$	/ k ₃ / k ₂ / k ₇ / k ₃ /	
		44	ŀ
		_	
		הכנסה דוגמא	
	חר זה תאים עד שמוצאים תא ריק	בזנון וווכנסוו, בווונים בזוו א	
	Å Å&A&A	<u>&</u> _	
	Ŷ ĠŶŶĦŶ	7 -	
	U		
_		40	
		46)

	VIOLET 20123
	הכנסה דוגמא בזמן ההכנסה, בוחנים בזה אחר זה תאים עד שמוצאים תא ריק
	בותן וווכנטון, בוונים בוו אווו זוו ואים עד שנובאים ואיז יון
_	A 3. A & _ A & _
	<u> </u>
	בדיקה מספר ס
	U
	47
	הכנסה דוגמא
_	בזמן ההכנסה, בוחנים בזה אחר זה תאים עד שמוצאים תא ריק
	ART PAR
	בדיקה מספר 1 U
	48
	_
	הכנסה דוגמא
	בזמן ההכנסה, בוחנים בזה אחר זה תאים עד שמוצאים תא ריק
	ART PAR
	Ţ
	בדיקה מספר 2
	U .

הכנסה דוגמא

בזמן ההכנסה, בוחנים בזה אחר זה תאים עד שמוצאים תא ריק



בדיקה מספר 3

50

שיטת המיעון הפתוח

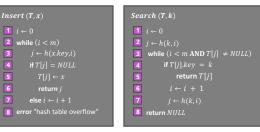
$$h: U \times \{0,1,\dots,m-1\} \to \{0,1,\dots,m-1\}$$
Universe coef native index (probe number)

תא שנגיע אליו בבדיקה ה-i-ית (בהנחה שבבדיקות הקודמות הגענו לתאים תפוסים)

אנז דורשים שסדרת התאים הנבדקים h(k,0),h(k,1),...,h(k,m-1) תהיה תמורה של אנו דורשים שסדרת התאים הנבדקים של דבר נבחן את כל תאי הטבלה $\{0,1,...,m-1\}$

51

הכנסה וחיפוש בשיטת המיעון הפתוח



 מחיקות בשיטת המיעון הפתוח	
 לא ניתן פשוט לסמן את התא כתא ריק על ידי הצבת NULL אם נעשה זאת לא נוכל לשלוף מפתח אשר במהלך הכנסתו נבדק תא זה ונמצא תפוס 	
- DELETED - • פתרון אפשרי: לסמן את התא על-ידי הצבת הערך המיוחד	
• בהכנסה להתייחס לתא כזה כאל תא ריק	
• בחיפוש לעבור על התא מבלי לעצור	
- חסרון: זמני ריצה אינם תלויים עוד במקדם העומס $lpha$	
- כאשר יש למחוק איברים, בוחרים בדרך כלל בשיטת השרשור	
•	53
	55
_	
- מיעון פתוח – איך מחפשים תא פנוי?	
-	
בדיקה לינארית (linear probing)	
- גיבוב כפול (double hashing)	
-	
•	
-	
_	
	54
 שיטת המיעון הפתוח	
$h: U \times \{0,1,\ldots,m-1\} \rightarrow \{0,1,\ldots,m-1\}$	
Universe מספר הבדיקה index (probe number)	
– תא שנגיע אליו בבדיקה ה- i -ית (בהנחה שבבדיקות הקודמות הגענו לתאים $h(k,i)$ •	
תפוסים) - אנו דורשים שסדרת התאים הנבדקים $h(k,0), h(k,1),, h(k,m-1)$ תהיה תמורה של -	
סבלה את כל תאי הטבלה (0, 1,, $m-1$) כך שבסופו של דבר נבחן את כל האי	
- מקדם העומס $lpha \leq 1$	

Linear Probing בדיקה לינארית
בזמן ההכנסה, אם התא תפוס נבדוק האם התא הבא בטבלה ריק וכך נמשיך
עד למציאת מקום ריק.
HHULLALA
56
Linear Probing בדיקה לינארית
בזמן ההכנסה, אם התא תפוס נבדוק האם התא הבא בטבלה ריק וכך נמשיך עד למציאת מקום ריק.
ער ינוב אונינוןים ו־קן.
0 × A.S. X • &
4444
VVV
57
_
בדיקה לינארית Linear Probing
$h(k,i) = (h'(k) + i) \bmod m$
$n(\mathbf{x},t) = n(\mathbf{x},t) + t$ ווויטט m - $n'(\mathbf{x})$ - $n'(\mathbf{x})$
. אינדקס של התא הראשון שנבדק $h'(k) ight. ^{-1}$
• התאים נבדקים לפי סדר לינארי של אינדקסים (באופן מעגלי)
h'(k), h'(k) + 1, h'(k) + 2,

	בדיקה לינארית Linear Probing Linear Probing יש להכניס את המפתחות $m=1$ 18,41,22,44,59,32,31,73,19 לטבלת ניבוב בנודל $m=1$ $m=1$ פונקצית גיבוב $m=1$ $h'(k)=k \mod 1$	0	
59	בדיקה לינארית Linear Probing Linear Probing יש להכניס את המפתחות $m=1, 18,41,22,44,59,32,31,73,19$ לטבלת גיבוב בגודל $m=13$ באמצעות שיטת הבדיקה הלינארית עם פונקצית גיבוב $m=13$ $h'(k)=k \mod 13$	0	
81	בדיקה לינארית Linear Probing Linear Probing $rac{ extbf{vision}}{ extbf{vision}}$ יש להכניס את המפתחות $18.41,22,44,59,32,31,73,19$ לטבלת $18.41,22,44,59,32,31,73,19$ לטבלת בנודל $18.41,22,44,59,32,31,73,19$ פונקצית גיבוב $18.41,22,44,59,32,31,73,19$ פונקצית גיבוב $18.41,22,44,59,32,31,73,19$ פונקצית גיבוב $18.41,22,44,59,32,31,73,19$	0	

62	בדיקה לינארית Linear Probing בדיקה לינארית Linear Probing יש להכניס את המפתחות 18,41,22,44,59,32,31,73,10 לטבלת בגודל $m=1$ באמצעות שיטת הבדיקה הלינארית עם פונקצית ניבוב $m=1$ $k'(k)=k \mod 1$	0	
63	בדיקה לינארית Linear Probing Linear Probing דוגמה: יש להכניס את המפתחות 18,41,22,44,59,32,31,73,19 לטבלת גיבוב בנודל $k'(k) = k \mod 13$ פונקצית ניבוב $k'(k) = k \mod 13$	0	
64	בדיקה לינארית בדיקה לינארית Linear Probing דומה: יש להכניס את המפתחות $13,41,22,44,59,32,31,73,19$ לטבלת ניבוב בגודל $m=13$ באמצעות שיטת הבדיקה הלינארית עם פונקצית ניבוב $m=13$ $m=13$	0	

 בדיקה לינארית Linear Probing	
 2 41	
 ש להכניס את המפתחות 18,41,22,44,59,32,31,73,19 לטבלת	
 6 44 7 59 h'(k) = k mod 13 6 7	
 8 32 9 22	
 10 31 Seurch(S1) 11 73 12 19	
	65
 בדיקה לינארית	
 - קל לממש	
 י סובלת מהצטברות ראשונית (primary clustering)	
• נוצרים רצפים ארוכים של תאים תפוסים, המאריכים את זמן החיפוש	
	66
 גיבוב כפול	
 $n(k,t) = (h_1(k) + th_2(k)) \mod m$ בווי אוי אוי אוי אוי אוי אוי אוי אוי אוי	
 אינדקס של התא הראשון שנבדק $h_1(k)$ -	
 $h_2(k)$ - לאחר מכן נבדקים תאים שמיקומיהם רחוקים זה מזה במרחק של k	
	67

גיבוב כפול 2 3 4 5 6 7 8 9 יש להכניס את המפתחות 18,41,22,44,59,32,31,73,19 לטבלת 18 עם הכפול הכפול שיטת בגודל m=13 גיבוב בגודל $h_1(k) = k \bmod 13$ פונקציות גיבוב $h_2(k) = 1 + k \bmod 11$ 22 11 12 68 גיבוב כפול יש להכניס את המפתחות 18,41,22,44,59,32,31,73,19 לטבלת 18 44 5 6 7 8 גיבוב בגודל m=13 באמצעות שיטת הגיבוב הכפול עם **3**2 $h_1(k) = k \bmod 13$ פונקציות גיבוב $h_2(k) = 1 + k \bmod 11$ $h_2(32) = 11$ 22 69 גיבוב כפול 4 5 6 7 32 18 יש להכניס את המפתחות 18,41,22,44,59,32,31,73,19 לטבלת 31 גיבוב בגודל m=13 באמצעות שיטת הגיבוב הכפול עם 44 59 $h_1(k) = k \bmod 13$ פונקציות גיבוב $h_2(k) = 1 + k \bmod 11$ 22 70

		0 1 2 41	
דוגמה:		3	
יש להכניט את המפתחות $18,41,22,44,59,32,31,73,19$ ויש להכניט את המפתחות $m=1$ באמצעות שיטת הגיבוב הכפול עם $m=1$ פונקציות גיבוב 11 השול $h_1(k)=k \bmod 13$ פונקציות $h_2(k)=1+k \bmod 11$	לת 19 $h_2(19) = 9$	4 32 5 18 6 44' 7 59 8 73 9 22 10 1 31	

71

גיבוב כפול

יש להכניס את המפתחות 18,41,22,44,59,32,31,73,19 לטבלת גיבוב בגודל m=13 באמצעות שיטת הגיבוב הכפול עם $h_1(k) = k \bmod 13$ פונקציות גיבוב $h_2(k) = 1 + k \bmod 11$

72

גיבוב כפול

- עבור שני מפתחות שמתחילים בדיקה מאותו תא, הצעד יכול להיות שונה
 - יש להבטיח שחיפוש יסרוק את טבלת הגיבוב כולה •
 - m חייב להיות זר לגודל הטבלה של הערך $h_2(k)$
 - יזוגי מספר אי-זוגי $m=2^p$ פר שתמיד הפיק אי-זוגי $m=2^p$
- יובי מספר שלם חיובי העיק מספר שלם חיובי h_2 את אשוני ולבנות את mm-קטן מ





 ניתוח של שיטת הגיבוב הפתוח	
· <mark>הנחה</mark> : גיבוב אחיד (uniform hashing) בהינתן מפתח כלשהו k , ההסתברות של כל אחת מ-! m התמורות של $\{0.1,,m-1\}$	
$rac{1}{m!}$ להיות סדרת הבדיקות ל k היא	
\cdot בדיקה לינארית - $\Theta(m)$ סדרות שונות \cdot גיבוב כפול - $\Theta(m^2)$ סדרות שונות \cdot גיבוב כפול - $\Theta(m^2)$	
•	
	7.4
	74
 ניתוח של שיטת הגיבוב הפתוח	
משפט: תוחלת מספר הבדיקות הנערכות בעת הכנסת איבר היא $\frac{1}{n-1}$, בהנחת	
$^{1-lpha}$ הגיבוב האחיד.	
. אם $\frac{1}{2}=2$ (טבלת הגיבוב חצי מלאה), $2=\frac{1}{1-lpha}=2$	
אם $\alpha=0.9$ (טבלת הגיבוב 90% מלאה), אם $\alpha=0.1$ מלאה), מראה), אם פרוע מעלת הגיבוב 10% מלאה), או	
אם $lpha$ מתקרב ל-1, מספר הבדיקות שואף לאינסוף $lpha$	
•	
	75
משפט: תוחלת מספר הבדיקות הנערכות בעת הכנסת איבר היא $\frac{1}{r_{-n}}$, בהנחת	
$^{1-lpha}$ הגיבוב האחיד.	
• הסבר אינטואיטיבי:	
• מספר הבדיקות עד לתא הפנוי הראשון ≈ התפלגות גיאומטרית עם הסתברות להצלחה $1-lpha$	
$rac{1}{1-lpha}$ תוחלת מספר הבדיקות היא $rac{1}{1-lpha}$.	
•	
	70
	76

ניתוח זמני ריצה של טבלת גיבוב $\alpha = \frac{n}{m}$ מקדם העומס (load factor) $\alpha = \frac{n}{m}$ מדרם העומס (load factor) $\alpha = 0.(1)$ $\alpha =$

77



78

גיבוב אוניברסלי

עיון־

- בחירת פונקציית גיבוב באופן אקראי, בדרך שאינה תלויה במפתחות שיאוחסנו בטבלת הגיבוב
 - הרעיון: לבחור פונקצית גיבוב באופן אקראי מתוך מחלקה ${\mathcal H}$ של פונקציות שתוכננה מראש.
- הבחירה האקראית מבטיחה שלא קיים קלט יחיד שעבורו התנהגות האלגוריתם היא תמיד הגרועה ביותר

 - גיבוב אוניברסלי הגדרה	
$\{0,1,,m-1\}$ קבוצה סופית של פונקציות גיבוב מ- U אל התחום $\{0,1,,m-1\}$ תיקרא קבוצה $rac{f k}{M}$ אם עבור כל זוג מפתחות שונים ${\cal R}$ ${\cal R}$ תיקרא קבוצה $rac{f k}{M}$ ורך ${\cal R}$ ווניברסלית ${\cal R}$ של פונקציות ניבוב.	
דוגמא לקבוצה אוניברסלית של פונקצית גיבוב \mathbf{r} ביבוב כתובת \mathbf{r} מו מורכבת מ-22 ביטים \mathbf{r} מתובת \mathbf{r} מו מורכבת מ-22 ביטים \mathbf{r} בינון לראות כתובת \mathbf{r} כרביעיה \mathbf{r} (x_1,x_2,x_3,x_4), x_1,x_2,x_3,x_4 מקבל ערכים בין \mathbf{r} ל-255 \mathbf{r} כנבות \mathbf{r} לראות ראשוני \mathbf{r} ביבות \mathbf{r} היות ראשוני \mathbf{r} \mathbf{r} סידו \mathbf{r} \mathbf{r} \mathbf{r} סידו \mathbf{r} \mathbf{r} \mathbf{r} סידו \mathbf{r}	

טבלת גיבוב **Hash Table**

82

81