הגדרות

אלגוריתם - דרך שיטית וחד משמעית לביצוע של משימה מסוימת במספר **סופי** של צעדים

פסאדו קוד – תיאור מצומצם ולא רשמי לאלגוריתם של תוכנית מחשב

זמן ריצה – מספר פעולות היסוד המבוצעות על תוכנית כלשהי למה פעולות יסוד ולא זמן? על מנת להתעלם מאספקטים טכנולוגים כמו סוגי מכונות

דוגמא: חיפוש בינארי מול חיפוש טרינרי

חיפוש טרינרי	חיפוש בינארי
$T(n) = \begin{cases} T\left(\frac{n}{3}\right) + 4, n \ge 2\\ 1, n \le 2 \end{cases}$	$T(n) = \left\{ T\left(\frac{n}{2}\right) + 2, n \ge 2 \right\}$
$(1, n \le 2)$	$(1, n \leq 2)$
$T(n) = \dots = T\left(\frac{n}{3^k}\right) + 4 \cdot k = 4 \cdot \log_3 n$	$T(n) = \dots = T\left(\frac{n}{2^k}\right) + 2 \cdot k = 2 \cdot \log_2 n$

$$2 \cdot \log_2 n = 2 \cdot \frac{\log_3 n}{\log_3 n} = 3.17 \cdot \log_3 n < 4 \cdot \log_3 n$$

אסימפטומטיקה – הערכה של קצב גידול של פונקציה

הסם עליון – נאמר ש-
$$f(n) \in O(g(n))$$
 אם:
$$\exists c>0, n_0 \geq 0: \forall n>n_0 \qquad f(n) \leq c \cdot g(n)$$

הסם תחתון – נאמר ש-
$$f(n) \in \Omega(g(n))$$
 אם:
$$\exists c>0, n_0 \geq 0: \forall n>n_0 \qquad f(n) \geq c \cdot g(n) \geq 0$$

הסם הדוק – נאמר ש-
$$f(n)\in \theta(g(n))$$
 אם:
$$\exists c_1,c_2>0,n_0\geq 0: \forall n>n_0 \qquad c_1\cdot g(n)\leq f(n)\leq c_2\cdot g(n)$$

סימון	שם
$O(c^n)$	אקספוננציאלי
$O(n^c)$	פולינומי
$O(n \cdot log(n))$	
O(n)	לינארי
$O(\sqrt{n})$	
$O(\log(n))$	לוגריתמי
$O(\log(\log(n)))$	
0(1)	קבוע
$O\left(\frac{1}{n}\right)$	

מיונים

הקדמה:

מיון מבוסס השוואות – מיון מבוסס השוואות מסדר אלמנטים במערך ע״י השוואה, בדרך כלל ע״י האופרטורים {≤,≥}

כל אלגוריתם המיון המבוסס על פעולות השוואה דורש לפחות $\Omega(n \cdot \log(n))$ פעולות השוואה במקרה במקרה הגרוע ביותר

הוכחה: עבור מיון n איברים יש צורך בעץ החלטה עם n! עלים (עבור כל אחת מהפרמוטציות השונות), בעץ החלטה בגובה n יש לכל היותר 2^h עלים, ולכן:

$$2^h \ge n! \Rightarrow h \ge \log_2(n!) \in \Omega(n \cdot log n)$$

על מנת להשיג סיבוכיות מיון טובה יותר מ- $\Omega(n \cdot logn)$ (ליניאריים) נוכל לאבד את הכלליות ולהתעסק על מנת להשיג סיבוכיות מיון טובה יותר מ- $Counting\ Sort$), סדרה של מספרים הנמצאים בטווח חסום (Radix Sort)

			וססי השוואות	לא מב			
	Bubble	Selection	Insertion	Merge Sort	Quick Sort	Radix Sort	Counting
	Sort	Sort	Sort				Sort
T(n)	T(n)			T(n)	T(n)		
	= (n-1)			$= 2 \cdot T\left(\frac{n}{2}\right)$	=T(k)		
	+ ··· + 1			2 (2)	+T(n-k-1)		
	n-1			+ O(n)	+ O(n)		
	$=n\cdot\frac{n-1}{2}$						
0	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n \cdot logn)$	$O(n^2)$	$O(d \cdot (n+k))$	O(n+k)
Ω	$\Omega(n^2)$	$\Omega(n^2)$	$\Omega(n)$	$\Omega(n \cdot logn$	$\Omega(n \cdot logn$	$\Omega(d \cdot (n+k))$	$\Omega(n+k)$
Θ	$\Theta(n^2)$	$\Theta(n^2)$		$\Theta(n \cdot logn)$	$\Theta(n \cdot logn)$	$\Theta(d \cdot (n+k))$	$\Theta(n+k)$
					בהסתברות		
					גבוהה		

מבני נתונים

סיבוכיות מחיקה	סיבוכיות חיפוש	סיבוכיות הכנסה	מוטיבציה	הגדרה	מבנה נתונים
0(1)	O(n)	0(1)	גודל המערך ידוע בזמן קומפילציה • רוצים Random • Access (גישה ישירה בזמן קבוע)	מבנה שמורכב מאוסף של תאים סדרתיים בזיכרון בעל גודל קבוע	מערך סטטי
0(1)	O(n)	עד כדי $ heta(1)$ עד ריפ i ze $ heta(n)$	גודל המערך לא ידוע בזמן קומפילציה	מבנה שמורכב מאוסף של תאים סדרתיים בזיכרון בעל גודל ניתן לשינוי	מערך דינמי
O(n)	O(logn)	בגלל $O(n)$ בגלל shifting האיברים	נרצה לבצע חיפוש מהיר, כאשר יש יותר חיפושים מאשר הכנסות	אוסף של תאים סדרתיים בזיכרון כך שכל איבר קטן מהאיבר הבא לפי אופרטור השוואה כלשהו { ,≥,}	מערך דינמי ממויין
0(1)	O(n)	0(1)	יותר הכנסות או אין צורך ב- Random Access, פחות אפקטיבי (זניח) מבחינת זיכרון עבור שמירה גם על גודל המצביע	 לכל איבר יש ערך לכל איבר יש לכל היותר מצביע יחיד לאיבר הבא 	רשימה מקושרת
O(n)	O(n)	O(n)	 מימוש לתור עדיפויות היות ומחיקה מהראש ומהסוף היא פעולה קלה מחיקה של איברים זהים כאשר אין זיכרון לטעון ל-HashSet 	 לכל איבר יש ערך לכל איבר יש לכל היותר מצביע יחיד לאיבר הבא, שהוא קטן מהאיבר הבא לפי אופרטור השוואה כלשהו {,≤,≥} 	רשימה מקושרת ממוינת
0(1)	-	0(1)	LIFO •	• מבנה נתונים אשר תומך ב סדר פעולות היסטורי (האחרון הוא הראשון לצאת)	מחסנית
0(1)	-	0(1)	• FIFO הערה: ניתן לממש תור בעזרת 2 מחסניות ומחסנית בעזרת 2 תורים (יש שקילות)	מבנה נתונים אשר תומך ב סדר פעולות היסטורי (האחרון הוא האחרון לצאת)	תור

ע ע ער בוצרי (לכל קודקוד יש ער ולכל היותר 2 בינארי (מות היותר של היותר 2 בינארי (מות היותר של היותר 2 בינארי (מותר של היותר 2 בינארי מותר של מותר מיותר מיותר 2 בינארי מותר של מותר מיותר 2 בינארי מותר של מיותר 2 בינארי מותר של מותר 2 בינארי מותר של מותר 2 בינארי מותר 3 בינארי מותר 2 בינארי מותר 3 בינארי מותר 2 בינארי מותר 3 בינאר מותר 3 בינאר מותר 3 בינאר מות				I			
בינארי (קודקודים) כך שעבור כל קודקוד בעל במרת עך השמאלי מר. במרת עך השור ממונים, מברם ממונים ממונים, מברם ממונים מברצעו הרבה השנויים במות המנוסה או מחיקה. בייה של עך מרונים מות במצוים הברם מים ממושע עדי אופרטור ממות שרי מערך כך שהבן שוה ביחס לאפורטור מברב מברם ערמה הוא (ח) 0 כי משר ראש הערמה במות ממולים אשר הואם רומבן בעדיפות לפי מברב מברם על מות לאפשר חיפוש Heapify שוה ביחס לאפרטור ממותש עדי מערך כך שהבן בוב מות שר אופרטור מברב מברם ערמה הוא (ח) 0 כי משר ראש הערמה במבר הוא (ח) 0 כי משר ראש הערמה במברם הוא מומבר בברם עד מברם במברם מברם מברם מברם מברם מברם מבר	O(n)	O(n)	O(n)				
ע ע עריפוש בינארי כך שכמות הרמות (
עץ עו איוו שור בירים הגדולים מ-יצ. בעל עץ חיפוש בינהר יכך שכמות המיבת עץ הימיני של לוגריתמית ביחס לכמות האיברים של לוגריתמית ביחס לכמות האיברים של לוגריתמית ביחס לכמות האיברים מיפוש של לוגריתמית ביחס לכמות האיברים מיפוש היו מוכחות עבור גובה עץ יותר ביחס לוגרית מית ווען איזום שחור שבור גובה עץ יותר ביחס לוגרית מית ביחס לכמות הרפושים גדולה, אך נעדיף להשתמש בו אם ממות הכנסה או מחייקה, מותר "סלחמי" ועדיף עד שחור שהוא יותבצע הרבה רוטעות לוגרית לוגרית ליותר "סלחמי" ועדיף עץ אדום שחור שהוא יותבצע הרבה רוטעות לוגרית לוגרים לוגר					ם) כך שעבור כל קודקוד בעל	קודקודי	בינארי
ע ע אר יפוש בינארי כך שכמות הרמות ע ע אר יפוש בינארי כך שכמות הרמות או לוגריתמית ביחס לכמות האיברים ע אר יחיפוש של לוגריתמית ביחס לכמות האיברים ע ארום שחור ביחס לכמות האיברים אינוי ש ארום שחור עבור גובה ע ארום שחור שהוא יתבצעו הרבה רוטציות ולכן ממות הרמש בו אם מתבצעים הרבה שינויים למות הרמשם ביחס להשואה עית אופרטור בית ערמה הוא (10) עבור שם הארום ארום שחור שהוא איז שופרטור בית ערמה הוא (10) עבור שם ארום שחור שהוא איז שופרטור בית בית אופרטור בית ממומש ע"י מערך כך שהבם מתובים ארול אופרטור בית						:x ערך	
ערמה ער חיפוש בינארי כך שכמות הרמות מיל לגריתמית ביחס לכמות האיברים בעץ יותר לגריתמית ביחס לכמות האיברים מיל לגריתמית ביחס לכמות האיברים בעץ יותר לגריתמית ביחס לכמות האיברים מילון מעץ אדום שחור שבר הגבה עץ יותר לגריתמים ביחס מונים, מבנה נתונים מאוזן מעץ אדום שחור שבר הגבה עץ יותר לגריתמים ביחס מונים, מבנים מילון מעץ אדום שחור שהוא יותר ייחס מונים מונים מילום מיל אדום שחור שהוא יותרי של אדום שחור שהוא מייחס מונים ליסוף, מבר מילום מיל לק ווקדון קטן או ביחס לאופרטור שוה ביחס לאופרטור ביחס לאופרטור ביחס לאופרטור ביחס מונים ען יותר מומש עייי מערך כך שהבן שווה ביחס לאופרטור ביחס לאופרטור ביחס לאופרטור ביחס מומש עייי מערך כך שהבן מומש עייי מערך כך שהבן והבן הימני באינדקס 1 + 2 בייחס של 1 ביחס מבנה נתונים אשר תומך בעדיפות לפי באינדקס 0 + 2 ביחס מבנה נתונים אשר תומך בעדיפות לפי באינדקס 0 + 2 ביחס מבנה נתונים אשר תומך בעדיפות לפי באינדקס 0 + 2 ביחס מבנה נתונים אשר תומך בעדיפות לפי באינדקס 0 + 2 ביחס מבנה נתונים אשר תומך בעדיפות לפי באינדקס 1 + 2 ביחס מבנה נתונים אשר תומך בעדיפות לפי באינדקס 1 + 2 ביחס מבנה נתונים אשר תומך בעדיפות לפי ביחס מבנה נתונים אשר תומך בעדיפות לפי ביחס מבנה ביחס מבנה נתונים אשר תומך בעדיפות לפי ביחס מבנה נתונים משר תומך בעדיפות לפי ביחס מבנה נתונים משר תומך בעדיפות לפי ביחס מבנה ביחס מבנה נתונים משר תומך בעדיפות לפי ביחס מבנה נתונים משר תומך בעדיפות לפי ביחס מבנה ביחס מב					x-הערכים הקטנים מ	.1	
ע עי חיפוש בינארי כך שכמות הרמות מורפות של הימני בינארי כך שכמות הרמות מורפות של הימני בינארי כך שכמות הרמות מורפות של הימני בינארי בינאר							
עץ עין חיפוש בינארי כך שכמות הרמות שור האיברים יעיל הייפוש של לוגריתמיים למות האיברים יעיל הייפוש של לוגריתמיים למות האיברים מחוכים, מבנה נתונים איברים ביע אדום שחור שבוא אדום שחור שבוא אדום שחור שבוא אדים ביע מות הכנסה או מחיקה, אדים שחור שבוא יותר של מת הבצעים הרבה שינויים במות החיפושים גדולה, אך עדיפות ערמה ביע מתנים ביע ביע ביע מתנים ביע ביע מתנים ביע					x-הערכים הגדולים מ	.2	
ייניל מאוזן בעץ אדום שחור בעץ ייניל אדום שחור אוגרתפיי ביחס לכמות האיברים וחלת עבור גובה עץ אדום שחור אינות בעור בורם עץ אדום שחור ביחס אומיקה, אדי מות בעור הרבה חיטציות ועץ אדום-שחור, מות החיפושים גדולה, אך נעדיף לא אדום שחור שהוא יתבצעו הרבה שינויים במות החיפושים גדולה, אך נעדיף עץ אדום שחור שהוא יתבצעו הרבה חיטציות ולכן כמות החיפושים גדולה, אך נעדיף עץ אדום שחור שהוא יתבצעו הרבה חיטציות ולכן כמות החיפושים גדולה אך נעדיף עץ אדום שחור שהוא יתבצעו הרבה חיטציות ולכן כמות החיפושים גדולה אך נעדיף עץ אדום שחור שהוא יתבצעו הרבה חיטציות ולכן במקום חיפושי ביחס אומרטור אופרטור אופרטור אופרטור שוה ביחס לאופרטור המותר של בנין מודים את הגדול ב-1 בנייה של עץ שיים על אופרטור הבן המני באינדקס 1 אופרטור בעדיפות לפי באינדקס 1 ליינים אופרטור בעדיפות לפי באינדקס 1 ליינים אופרטור בעדיפות לפי במנה נתונים אשר תומך בעדיפות לפי באינדקס 1 ליינים אופרטור בעדיפות לפי במנה נתונים אשר תומך בעדיפות לפי ביחס במנה נתונים אשר תומך בעדיפות לפי ביחס במנה ביחס לאופרטור בעדיפות לפי ביחס ביחס ביחס ביחס ביחס ביחס ביחס ביח					נמצאים בתת עץ הימני		
ייניל מאוזן בעץ אדום שחור בעץ ייניל אדום שחור אוגרתפיי ביחס לכמות האיברים וחלת עבור גובה עץ אדום שחור אינות בעור בורם עץ אדום שחור ביחס אומיקה, אדי מות בעור הרבה חיטציות ועץ אדום-שחור, מות החיפושים גדולה, אך נעדיף לא אדום שחור שהוא יתבצעו הרבה שינויים במות החיפושים גדולה, אך נעדיף עץ אדום שחור שהוא יתבצעו הרבה חיטציות ולכן כמות החיפושים גדולה, אך נעדיף עץ אדום שחור שהוא יתבצעו הרבה חיטציות ולכן כמות החיפושים גדולה אך נעדיף עץ אדום שחור שהוא יתבצעו הרבה חיטציות ולכן כמות החיפושים גדולה אך נעדיף עץ אדום שחור שהוא יתבצעו הרבה חיטציות ולכן במקום חיפושי ביחס אומרטור אופרטור אופרטור אופרטור שוה ביחס לאופרטור המותר של בנין מודים את הגדול ב-1 בנייה של עץ שיים על אופרטור הבן המני באינדקס 1 אופרטור בעדיפות לפי באינדקס 1 ליינים אופרטור בעדיפות לפי באינדקס 1 ליינים אופרטור בעדיפות לפי במנה נתונים אשר תומך בעדיפות לפי באינדקס 1 ליינים אופרטור בעדיפות לפי במנה נתונים אשר תומך בעדיפות לפי ביחס במנה נתונים אשר תומך בעדיפות לפי ביחס במנה ביחס לאופרטור בעדיפות לפי ביחס ביחס ביחס ביחס ביחס ביחס ביחס ביח	O(logn)	O(logn)	O(logn)	• הכנסה. מחיקה. חיפוש	ש בינארי כך שכמות הרמות	עץ חיפו	עץ
אוד בעץ אדום שחור אוד בעץ אדום שחור אוד אוד בעד איז אוד בעד בעד איז אוד בעד אוד בעד אוד בעד אוד בעד אוד בעד איז אוד בעד בעד בעד בעד בעד בעד בעד בעד בעד בע				_ · · · · · · · · · · · · · · · · · · ·	יתמית ביחס לכמות האיברים	של לוגר	
ארונים או אין				• חיפוש תווכים. איברים		בעץ	מאוזן
ערמה (logn) עובר השרעה בו אם מאוזן מעץ אדום שחור שהוא עץ יותר לוגרתמי מות הכנסה או מחיקה, אם מתבצעים הרבה רוטציות ולכן כמות החיפושים גדולה, אך נעדיף לא אחרו שהוא עדי אופרטור (מקום חיפוש שוה ביחס לאופרטור הערך של כל קודקוד קטן או להשואה ע"י אופרטור שוה ביחס לאופרטור שוה ביחס לאופרטור שוה ביחס לאופרטור (מות של באינדקס 1 לוצר של באינדקס 1 לוצר שהבן הימני באינדקס 1 לוצר שהבן הימני באינדקס 1 לוצר שהבן הובן הימני באינדקס 1 לוצר שהבן בודים ול מנת לאפשר חיפוש באינדקס 1 לוצר שהבן בודים ול מנת לאפשר חיפוש באינדקס 1 לוצר שהבן בודים ול מנת לאפשר חיפוש באינדקס 1 לוצר של מנקציית גיבוב בייה של עץ בודיפות לפי ביוניים של מנומים אשר תומך בעדיפות לפי ביונים של מנומים של מנות מנומים של מנומ					AVL	-	
ערמה ($O(logn)$ עובר השורש בו אם מתבצעים הרבה שימיים כמות החיפושים גדולה, אך עדיפות ערמה ($O(logn)$ עובר השורש ($O(logn)$ ערמה ($O(logn)$ ער						•	
ערמה ($O(logn)$ ערמה ($O(logn)$ ערמה ($O(logn)$ ממוש עיית של אופרטור (במוש ביחס לאופרטור ממוש שיי מערך עך שהבן שוה ביחס לאופרטור (ממוש שיי מערך עך שהבן שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i שוה ביחס לאופרטור) באופרטור (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i בענד (i שוה ביחס לאופרטור) באינדקס 1 + i בענד (i					-		
ערמה $O(logn)$ ערמדיש עריי ערך עו מפתח $O(logn)$ ערמה $O(logn)$ ערמה $O(logn)$ ערמים ערן ערן עו מפתח $O(logn)$ ערמה $O(logn)$ ערמה $O(logn)$ ערך עו מפתח $O(logn)$ ערמה							
ערמה $O(logn)$ עובר השורש עדי אדום שחור שהוא עדי אדום ערמה $O(logn)$ עובר השורש $O(logn)$ עובר השורש עדי אופרטור $O(logn)$ ערמה אוא פרטור $O(logn)$ ערמה אוא עושים אואפרטור $O(logn)$ ערמה אוא אופרטור $O(logn)$ ערמה אוא פרטור $O(logn)$ ערמה אוא פרטור $O(logn)$ ערמה אוא פרטור $O(logn)$ ערמה אוא עושים אואפרטור $O(logn)$ ערמה אוא עושים אואפרטור $O(logn)$ ערמה אואפרטור $O(logn)$ ערמה אואפרטור $O(logn)$ ערמה אואפרטור $O(logn)$ ערמה אואפרטור $O(logn)$ אואפרטור $O(logn)$ ערמה אואפרטור אואפרטור $O(logn)$ ערמה אואפרטור אואפרטור אואפרטור ערמה אואפרטור ערפער אואפרטור ערפער אואפרטור ערפער אוייייי ערייי אואפרטור ערפער אואפרטור ערפער אואפרטור ערפער אואפרטור ערפער ערפער אואפרטור ערפער אואפרטור ערפער ע							
ערמה • להשתמש בו אם מתבצעים הרבה רוטציות ולכן כמות הסנסה או מחיקה, אם מתבצעים הרבה רוטציות ולכן כמות הסנסה או מחיקה, יותר "סלחני" (במקום חיפוש להשוואה ע"י אופרטור • להייברים בעץ ניתנים • ערמה • הערך של כל קודקוד קטן או להשווא הע"ז אופרטור • ממומש ע"י מערך כך שהבן מהערך שווה ביחס לאופרטור • ממומש ע"י מערך כך שהבן הימני באינדקס 1 + 12 אופרטור • בייה של עץ (מונים אשר תומך בעדיפות לפי מובנה נתונים אשר תומך בעדיפות לפי באינדקס 0 (מוברים בעץ ניתנים • לכל ערך יש מפתח (מוברים בעץ ניתנים • לבל ערך יש מפתח (מוברים • בעל פונקציית גיבוב • לבל מוברים קטנה • לבל ערך ליש מפתח (מוברים בעץ ניתן להשתמש ב- עוברים בען ניתן להשתמש ב- על פונקציית גיבוב • בעל פונקציית גיבוב • לבל ערך ליש מפתח (מוברים בעף בעדים מוברים בען פונקציית גיבוב • לבל ערך ליש מפתח (מוברים בען בעדים מוברים בען פונקציית גיבוב • לבל ערך ליש מפתח (מוברים בעל פונקציית גיבוב • לבל ערך ליש מפתח (מוברים בעל פונקציית גיבוב טובה: עוברים בען מוברים בעל מוברים בעל פונקציית גיבוב טובה: עוברים בעל מוברים בעל							
ערמה • למות החיפושים גדולה, אך אם מתבצעים הרבה שינוים במות הכנסה או מחיקה, יתבצעו הרבה רוטציות ולכן נעדיף עץ אדום שחור שהוא יתבצעו הרבה רוטציות ולכן יתבעה הרבה רוטציות ולכן יתבעה הרבה רוטציות ולכן יתבעה הרבה רוטציות ולכן יתבעה שורה שוחר שהוא ייתבעה היים בעץ ניתנים • עץ עדיפות (100 עבור מהשואה ע"י אופרטור • הערך של כל קודקוד קטן או שוה ביחס לאופרטור • הערך של כל קודקוד קטן או שוה ביחס לאופרטור • ממומש ע"י מערך כך שהבן ממומש ע"י מערך כך שהבן והימני באינדקס 1 / 2 / 2 אופרטור יובן הימני באינדקס 1 / 2 / 2 אופרטור יובן הימני באינדקס 1 / 2 / 2 אופרטור יובן הימני באינדקס 1 / 2 / 2 אופרטור יובן הימני באינדקס 1 / 2 / 2 אופרטור יובן הימני באינדקס 1 / 2 / 2 יובן מת לאפשר חיפוש, אופרטור יובן מבנה נתונים אשר תומך בעדיפות לפי יובן מבלת הופים אופרטור יובן מבנה בתונים בייבים על מת לאפשר חיפוש, יובן בייבים בייבים על מת לאפשר חיפוש, יובן בייבים בייבים ע"ד מוספה ומחיקה בסיבוכיות יובן בייבים בייבים על מוד לאפשר חיפוש, יובן בייבים בייבים על מודיקה בייבים של עץ יובן מודיקה בייבים בייבים עדים בייבים על מוד לאפשר חיפוש, יובן בייבים בייבים על עץ יובן בייבים בייבים על מודיקה בייבים על מוד לאפשר חיפוש עייד בייבים על מודיקה בייבים בייבים על מודיקה בייבים בייבים בייבים על מודיקה בייבים בייבים בייבים בייבים ב							
ערמה $O(logn)$ עובר השורש או מחיקה, וועד "סלחני" ערמה $O(logn)$ עובר השורש או מחיקה, וועד "סלחני" ערמה $O(logn)$ ערמה ע"י אופרטור $O(logn)$ ערמה ע"י אופרטור $O(logn)$ ערמה הוא $O(logn)$ ערמה הוא $O(logn)$ ערמה $O(logn)$ ערמיע ערמעים ערמף ערמף ערמף ערמף ערמף ערמף ערמף ערמף							
ערמה • כל האיברים בעץ ניתנים • עץ עדיפות (100 אופרטור • כל האיברים בעץ ניתנים • עץ עדיפות (100 עבור השורש (100 עבור שביים הוא (100 עבור שביים הוא (100 עבור עבור שביים הוא (100 עבור עבור שביים (100 עבור עבור עבור עבור עבור עבור עבור עבור							
ערמה ((((ו ערמים בער							
ערמה • כל האיברים בעץ ניתנים • עץ עדיפות • עץ עדיפות • עק							
ערמה (השואה ע"י אופרטור (במקום חיפוש (מניסים לסוף, מלוניים אואה ע"י אופרטור (במקום חיפוש (מניסים לסוף, מרך של כל קודקוד קטן או (מואף ע"י אופרטור (ממומש ע"י מערך של בניו (מואף ע"י מערך על בניו (עושים ע"י מערך על ביו (מואף ע"י מערך על ביו (עושים ע"י מערך על ביו (עושים ע"י מערך על ביו (מואף בעדיפות לפי ע"י מערך על ביו (מואף ע"י מערך על ביו (מואף ע"י מערף ע"י מערף על ביו (מואף ע"י מערף על ביו (מואף ע"י מערף ע"י מ							
ערמה (ערמה							
להשוואה ע"י אופרטור (במקום חיפוש מכניסים לסוף, חיפוש לשהו (חיפוש לשהו ביחס לאופרטור (חיפוש ליים אופרטור) אופרטור (חיפוש ע"י מערך על בניו (חיפוש (חיפוש (חיפוש ע"י מערך על בניו (חיפוש (חיפוש (חיפוש (חיפוש ע"י מערך על בניו (חיפוש (חיפ	O(loan)	עורר העורע	O(logn)	עע עדיפות		•	ערמה
• הערך של כל קודקוד קטן או שופרטור (רובצה עדיפות לפי שוה ביחס לאופרטור שוה ביחס לאופרטור שוה ביחס לאופרטור (רובש של מהערך של בניו מהשרך של בניו מהשמלי באינדקס 1 (רובש מתר בידיפות לפי ממומש ע"י מערך כך שהבן השמאלי באינדקס 1 (רובש מתר בידיפות לפי ממומש ע"י מערך כך שהבן השמאלי באינדקס 1 (רובש מתר בידיפות לפי מומרים בידיפות לפי מומרים בידיפות לפי מומרים של עד של מתר בידיפות לפי מומרים בידיפות לפי מומרים של ממומש בידיפות לפי מומרים של מומר לאפשר חיפוש, שופרטור (רובש מתר בידיפות לפי מומרים בידיפות לפי מומרים בידיפות לפי מומרים של מומר לאפשר חיפוש, בידיפות לפי מומרים בידים ב		· ·		-		-	
שווה ביחס לאופרטור אופרטור שווה ביחס לאופרטור לאופרטור לאופרטור אופרטור לאופרטור לאופרטור לאופרטור לאופרטור לאופרטור ביחס לאופרטור לאופרטור פווא מהערך של בניו מערך כך שהבן הערבה בייה של עץ לייש מעדיי מערך כך שהבן הערבה הוא (ח) 0 כי באשר ראש הערבה מבנה נתונים אשר תומך בעדיפות לפי מבנה מומחיקה בסיבוכיות לו מבנה מבנה נתונים אשר מבנה נתונים לאופרטור מבנה מבנה מבנה מבנה מבנה מבנה מבנה מבנה							
 Heapify מהערך של בניו הערה: בנייה של עץ הערה: בנייה של עץ הערה: בנייה של עץ בודקים מי מהבנים הוא מומש ע"י מערך כך שהבן השמאלי באינדקס 2i + 1 ס מומש ע"י מערך כך שהבן הוא ערמה הוא (n) 0 כי עושים Heapify עושים Heapify עושים אובן הימני באינדקס 2i + 2 ס מגודל המערך מגודל המערך מגודל המערך מבנה נתונים אשר תומך בעדיפות לפי באינדקס 0 של מנת לאפשר חיפוש, על מנת לאפשר חיפוש, הוספה ומחיקה בסיבוכיות בעל פונקציית גיבוב בוב סובה: באינדם טובה: ס(1) באובה: ס(1) במות הפלטים קטנה ומחיקה בעובה: סובר מומר של ביים שונקציית גיבוב טובה: סובר מומר של ביים שונה של ביים של							
• ממומש ע"י מערך כך שהבן הערמה הוא (ח) כי רואם הוא עדי מערך כך שהבן בנייה של עץ רמה הוא (ח) כי רואם הוא בודקים מי מהבנים הוא (ח) כי רואם הערמה באינדקס (חיבות באינדקס (חיב		0 (11)		(110 13111			
אופים מי מהבנים הוא $O(n)$ כי $O(n)$ ערמה הוא $O(n)$ כי $O(n)$ עושים Heapify עושים Heapify כחבר הוהבן הימני באינדקס $O(n)$ עושים Heapify עושים $O(n)$ עושים $O($				הערה : בנייה של עע			
עם עדיפות ומחליפים Heapify עושים Heapify מגודל המערך מגודל המערך מגודל המערך מגודל המערך באשר ראש הערמה 0 מגודל המערך מבנה נתונים אשר תומך ב עדיפות לפי אופרטור שופרטור על מנת לאפשר חיפוש, על מנת לאפשר חיפוש, על מנת לאפשר חיפוש, דable: $0(1)$ Table: $0(1)$ 0 Direct Access Table $0(1)$							
אופרטור באינדקס ס מבנה נתונים אשר תומך ב עדיפות לפי מבנה נתונים אשר תומך ב עדיפות לפי אופרטור מבנה נתונים אשר תומך ב עדיפות לפי אופרטור שופרטור סבלת אופרטור פעדים מפתח על מנת לאפשר חיפוש, דable: חוספה ומחיקה בסיבוכיות הוספה ומחיקה בסיבוכיות קבועה, ניתן להשתמש ב- פונקציית גיבוב טובה: חוספה ומחיקה בסיבוכיות על חוספר מוספר מ							
מבנה נתונים אשר תומך ב עדיפות לפי אופרטור אופרטור Direct Access Table: Access Table: 0(1) Table: 0(1) Direct Access Table O(1) Direct Access Table O(1) Direct Access Table O(1) Direct Access Table O(1) Direct Access Table							
מבנה נתונים אשר תומך ב עדיפות לפי אופרטור Direct Access Table: O(1) Table: O(1) Direct Access Table Direct Access Table Direct Access Table Direct Access Table O(1) Table: O(1) Direct Access Table Direct Access Table Direct Access Table O(1) Table: O(1) Direct Access Table				•			
אופרטור Direct Access Table:					באנוקסס		
אופרטור Direct Access Table:					תונים אוער תומר ר עדיפות לפי	מרוה וו	
טבלת לכל ערך יש מפתח על מנת לאפשר חיפוש, על מנת לאפשר חיפוש, דable: Access דable: סבלת בעל פונקציית גיבוב סובה: ס(1) דable: ס(1) סובר איינו איינו להשתמש ב- סיבול של סובר מות הפלטים קטנה מות לאפשר חיפוש, איינו לאפשר חיפוש היינו לאפשר הי					-		
דעבוב • בעל פונקציית גיבוב • הוספה ומחיקה בסיבוכיות • בעל פונקציית גיבוב • הוספה ומחיקה בסיבוכיות • ס(1) • ס(1) • ספונקציית גיבוב טובה: • ס(1) • ספות הפלטים קטנה • סוות הפלטים קטנה • סיבות דמשות ב- • ספות הפלטים קטנה • ספות הפלטים קטות הפלטים קטנה • ספות הפלטים קטות הפלטים קטנה • ספות הפלטים קטות הפלטים קטנה • ספו	Direct Access	Direct	Direct Access	על מות לאפשר חיפוש.			טבלת
O(1) Table: $O(1)$ - פונקציית גיבוב טובה: קבועה, ניתן להשתמש ב $O(1)$ Direct Access Table על					•		
0(1) על Direct Access Table על .1							**
	- (-)		- (-)				
מרמות ההלמים מנת לקבל סיבוכיות זו ע"י Upen Addressing: מרמות ההלמים מנת לקבל סיבוכיות זו ע"י	Open Addressing:	, (-)	Open	מנת לקבל סיבוכיות זו ע״י	רי. פנווניוופיס בו קסנוי מכמות הקלטים		
O(n) Open Addressing: מקום כאשר U זה $O(U)$ מקום כאשר U מקום כאשר		Open					
Addressing: $O(n)$ ממות התנגשויות צריכה מרחב המפתחות 2.			_				
Separate $O(n)$ האפשריים, או לצמצם את להיות קטנה ככל	Separate	_					
Channing: Separate שאפשר הזיכרון ולפתור בעיות			Separate		·		
תלוי מימוש, מערך של Separate Channing: (Collision) קלה לחישוב .3		Separate					
רשימות מקושרות (Channing: תלוי מימוש, Channing: 4			תלוי מימוש,				
0(list length) תלוי מימוש, אך, משלמים על זה בכך מערך של 5.	O(list length)	תُלוי מימוש,	מערך של	אך, משלמים על זה בכך			
שיש למפתח שמבנים דינמאים הם יעילים רשימות מקושרות מערך של בעוד שמערך של עצי	בעוד שמערך של עצי	מערך של	רשימות מקושרות	שמבנים דינמאים הם יעילים			
רשימות חיפוש בינאריים $O(list\ length)$ מפזרת אחיד מבחינת זיכרון, בעוד שטבלת מפזרת אחיד	חיפוש בינאריים	רשימות	O(list length)	מבחינת זיכרון, בעוד שטבלת	6. מפזרת אחיד		
O(logn) מקושרות גיבוב היא בזבזנית, בנוסף בעוד שמערך של מקושרות 7.	O(logn)	מקושרות	בעוד שמערך של	גיבוב היא בזבזנית, בנוסף	7. חד כיוונית לעיתים		
אם אנחנו רוצים להתעסק עם עצי חיפוש O(list length)		O(list length)	עצי חיפוש	אם אנחנו רוצים להתעסק עם			
Load Factor פותר התנגשויות מידע ממוין, מציאת איברים בינאריים בעוד שמערך עם • •	Load Factorעם		בינאריים	מידע ממוין, מציאת איברים	פותר התנגשויות	•	
סמוכים, חיפוש תווכים, $O(logn)$ של עצי חיפוש ופונקציית גיבוב טובה,	ופונקציית גיבוב טובה,	של עצי חיפוש	O(logn)	סמוכים, חיפוש תווכים,			

נוכל לקבל (0(1) Amortized	בינאריים $O(logn)$	עם Load ופונקציית <i>Factor</i>	להציג מבנה נתונים ממויין אז נעדיף להשתמש בעצים לדוגמא	לדוגמא: Open Addressing (מציאת מקום חדש לאובייקט)	•	
	עם Load Factor ופונקציית גיבוב טובה, נוכל לקבל (O(1 Amortized	גיבוב טובה, נוכל לקבל (0(1) Amortized		Separate Chaining (מערך של רשימות מקשורות, עצי חיפוש, וכו׳)	•	