

## מטלה 14:

מטרת הפרויקט:

להתנסות במימוש של מבני נתונים ואלגוריתמים, בחקירת ביצועיהם בפועל אל מול ניתוח תאורטי של הסיבוכיות שלהם, ובהצגה אקדמית של התהליך והמסקנות.

תיאור הפרויקט:

הפרויקט הוא מעין "מיני-מחקר", בו תיערך השוואה בין שני פתרונות לבעיית מציאת ומיון  $k$  האיברים הקטנים ביותר מתוך האיברים נתונים.

הגדרת הבעיה -

הקלט: מערך בגודל  $n$  של מספרים שלמים  
הפלט:  $k$  האיברים הקטנים ביותר בסדר ממין

שני הפתרונות ביניהם נשווה הם:

פתרון א': בניית ערמת מינימום מהמערך הנתון ע"י קריאה ל- Build-Min-Heap, ואח"כ קריאה ל-Heap-Extract  $k$  פעמים.

פתרון ב': הרצת RANDOMIZED-Select על המערך הנתון למציאת האיבר ה-  $k$ -קטן ביותר וחלוקת המערך סביבו, ואח"כ מיון של  $k$  האיברים השמאליים באמצעות האלגוריתם מיון-מהיר (QUICKSORT).

- נבצע חישוב תיאורטי של מספר ההשוואות עבור שתי השיטות
- נבדוק מה מספר ההשוואות בפועל על המערכים האקראיים שנתונים במטלה
- נשווה בין בתוצאות

### חישוב תיאורטי:

שיטה 1:

1. יצירת ערימה ריקה –  $O(1)$

2. Heap-sort:

ביצוע  $n$  insert פעמים, ובכל פעם מכניסים את האיבר לתחתית הערימה, ומוודאים שהוא נמצא במקום כלומר שה"אבא" שלו יותר קטן ממנו. עוברים בלולאה על גובה הערימה ( $\log n$ ) ומוודאים שזה אכן מתקיים ע"י השוואות.

במקרה הגרוע:

המערך המקורי יהיה ממין מהגדול לקטן, ואז בכל הכנסה נצטרך להעלות את האיבר עד למעלה, כך שעבור האיבר ה- $i$  ( $n \geq i \geq 1$ ) נבצע  $\log(i)$

השוואות, אז נוסחת הנסיגה היא:

$$T(n) = \Theta(\log n + \log(n-1) + \dots + \log 2) = \Theta(\log(n!)) \#1 = \Theta(n \log n) \#2$$

אז מספר ההשוואות הוא  $(\log(n!))$ , ובסדר גודל  $(n \log n)$

#1 - חוקי לוגריתמים

#2 - קירוב סטרלינג

במקרה הטוב:

המערך כבר ממין ולכן בכל הכנסה תתבצע רק השוואה אחת, וזמן הריצה יהיה  $n$  וזה מספר ההשוואות

3. arrayOfMin:

קוראת  $k$  פעמים לשיטה remove

השיטה remove מוציאה את האיבר שנמצא בראש הערימה, ושמה במקומו את האיבר האחרון בערימה שהוא עלה, אז כמובן שהוא לא נמצא במקומו.

השיטה קוראת ל-heapify - כאשר רק הצומת שבראש העץ לא נמצאת במקומה, heapify מורידה את האיבר שבראש הערימה עד לתחתית, כלומר היא יורדת  $\log(n)$  רמות, ובכל רמה היא מבצעת 2 השוואות – אחת עם הבן

הימני ואחת עם הבן השמאלי, ולכן זמן הריצה (מספר ההשוואות) הוא  
 $O(\log n) = 3 \log n$   $O((n-k)\log^2 + \dots + \log(n)^2)$   
 #3 - k הוא קבוע

סה"כ:

במקרה הטוב:  $O(n) = O(\log(n) + n + 1)$

במקרה הגרוע:  $O(n \log n) = O(\log(n) + n \log(n) + 1)$

שיטה 2:

1. randomizeSelect:

שיטה רקורסיבית שבכל איטרציה פועלת ב-2 שלבים:  
 שלב א': קוראת לשיטה randomizePartition, שבוחרת מספר  
 רנדומלי  $rand$  ( $1 \leq rand \leq n$ ) מהמערך, ומעבירה אותו  
 למקום שלו - את כל האיברים שקטנים מימנו שמה משמאל  
 ל- $A[rand]$  ואת אלו שגדולים מימנו שמה מימין ל- $A[rand]$ ,  
 כך שבסוף יוצא ש- $A[rand]$  נמצא במקום ה- $i$ , ו- $A[rand]$  הוא  
 האיבר ה- $i$  בגודלו.  
 השיטה עוברת בלולאה אחת על כל איברי המערך ולכן  
 מספר ההשוואות הוא  $O(n)$ . (בדיוק)

שלב ב': לאחר שלב א' יתכנו 3 מצבים:

1.  $k=i$  ובעצם סיימנו
2.  $k > i$  וזה אומר שהאיבר במיקום ה- $k$  נמצא בתת-המערך  
 $[n, \dots, 1+i]$  אז נשלח את תת-המערך בצורה רקורסיבית  
 לשיטה randomizeSelect
3.  $i > k$  וזה אומר שהאיבר במיקום ה- $k$  נמצא בתת-המערך  
 $[n, \dots, 1+A[i]]$  אז נשלח את תת-המערך בצורה רקורסיבית  
 לשיטה randomizeSelect

זמני הריצה בעצם תלויים באיבר הרנדומלי שהשיטה randomizePartition  
 בחרה, נבדוק מהן האפשרויות השונות:

במקרה הגרוע:

$K=1$ , ובשלב א' האיבר הרנדומלי שנבחר הוא במיקום ה-1 והאיבר ה- $k$   
 בגודלו נמצא בתת המערך  $A[1, \dots, n-1]$ , וזה המצב בכל איטרציה, אז מספר  
 ההשוואות הוא:

$$T(n) = c \cdot n + T(n-1) = O(n^2)$$

במקרה הטוב:

בשלב א', על הבחירה הראשונה, (או אחרי מספר בסדר גודל קטן מ- $n$  מצאנו  
 את האיבר שמיקומו הוא  $k$   
 מספר ההשוואות יהיה  $O(n)$

במקרה הממוצע:

בשלב א' האיבר הרנדומלי שנבחר נמצא במיקום ה- $\frac{n}{2}$  וכך קורה בכל  
 איטרציה, ואז ללא תלות בערך של  $k$  מספר ההשוואות הוא:  
 $(n \log n)T(n) = c \cdot n + T(\frac{n}{2}) = O$

2. Quicksort:

בשלב זה  $k$  האיברים הכי קטנים במערך נמצאים ב- $k$  המקומות הראשונים, נפעיל את השיטה quicksort על תת המערך  $A[1,...,k]$  שזמן הריצה שלה (מספר ההשוואות) הוא,  $O(k \log k)$  כך שא הוא קבוע ולכן זמן הריצה הוא  $O(1)$

3. arrayOfMin: (בשיטה זו לא נוספות השוואות)  
 השיטה מעתיקה את תת המערך  $A[1,...,k]$  (שעתה כבר ממורן) למערך חדש בגודל  $k$ , זמן הריצה הוא  $O(k)$ .  $K$  הוא קבוע ולכן זמן הריצה הוא  $O(1)$ .

## לסיכום:

מספר ההשוואות של שיטה 1:

במקרה הטוב:  $O(n)$

במקרה הגרוע:  $O(n \log(n))$

מספר ההשוואות של שיטה 2:

במקרה הגרוע:  $O(n^2)$

במקרה הטוב:  $O(n)$

במקרה הממוצע:  $O(n \log n)$

## סיכום תוצאות ההרצה + תאוריה:

### שיטה 1:

\*מצפים לקבל:

$$\log(n!) + 1 + (2 \cdot \log(n) + \dots + 2 \cdot \log(n-k)) = n \log(n) + 1 + k(\log(n) + \log(n-k)) \quad \text{במקרה הגרוע:}$$

$$n + 1 + (2 \cdot \log(n) + \dots + 2 \cdot \log(n-k)) = n + 1 + k(\log(n) + \log(n-k)) \quad \text{במקרה הטוב:}$$

גודל המערך					
	1000	500	200	100	
מספר השוואות	1160 3160 2394	644 1493 1258	323 583 500	206 306 296	8
	1995 3995 3071	1391 2240 1945	944 1204 1086	715 815 780	50
	2982 4982 4023	1965 3114 2665	1629 1889 1747	765 1328 1211	100

המקרה הטוב  
 המקרה הגרוע  
 תוצאת הרצה

### שיטה 2:

\*מצפים לקבל:

במקרה הטוב:  $n + k \log k$

במקרה הגרוע:  $k \log k + \frac{n(1+n)}{2} = (k)k \log k + n + (n-1) + \dots + 1$

במקרה הממוצע:  $k \log k + n \log n$

גודל המערך					8	א איברים מינימליים
1000	500	200	100			
1,024	524	224	124			
9,989	4,506	1,552	688			
500,524	125,274	20,124	5,074			
1894	1618	538	182			
1,282	782	482	238	50		
10,247	4,764	1,810	946			
500,782	125,532	20,382	5,332	100		
4233	1458	1159	757			
2,164	1,164	864	764			
10,629	5,146	2,192	1,328			
501,164	125,914	20,764	5,714			
5497	3726	2250	1412			

המקרה הטוב  
המקרה הממוצע  
המקרה הגרוע  
תוצאת הרצה

## תוצאות הרצת המערכים לדוגמה:

k=100	k=50	k=8	
<pre>Mmn14.main({ }); Select the array size 100 Choose how few minimal organs you want to print 100 In what way would you like to insert the array member? For select yourself, press - 1  For random selection press - 2 2 The original array: [875,89,156,876,940,292,408,147,683,212,88,117,717,4']  Method 1: The array of 100 minimal organs: [2,42,43,44,49,50,51,52,59,88,89,91,102,117,117,127,']  The number of comparisons made by method 1 is: 1211  Method 2: The array of 100 minimal organs: [2,42,43,44,49,50,51,52,59,88,89,91,102,117,117,127,']  The number of comparisons made by method 2 is: 1412</pre>	<pre>Mmn14.main({ }); Select the array size 100 Choose how few minimal organs you want to print 50 In what way would you like to insert the array members? For select yourself, press - 1  For random selection press - 2 2 The original array: [831,158,963,717,136,998,622,730,155,901,872,25,377,198,39]  Method 1: The array of 50 minimal organs: [2,3,11,13,25,38,44,67,70,83,112,115,118,127,129,134,136,1]  The number of comparisons made by method 1 is: 780  Method 2: The array of 50 minimal organs: [2,3,11,13,25,38,44,67,70,83,112,115,118,127,129,134,136,1]  The number of comparisons made by method 2 is: 757</pre>	<pre>Mmn14.main({ }); Select the array size 100 Choose how few minimal organs you want to print 8 In what way would you like to insert the array members? For select yourself, press - 1  For random selection press - 2 2 The original array: [805,115,525,467,619,654,310,663,730,245,681,446,584,329,]  Method 1: The array of 8 minimal organs: [26,36,39,113,113,115,121,123]  The number of comparisons made by method 1 is: 296  Method 2: The array of 8 minimal organs: [26,36,39,113,113,115,121,123]  The number of comparisons made by method 2 is: 182</pre>	n= 100
<pre>Mmn14.main({ }); Select the array size 200 Choose how few minimal organs you want to print 100 In what way would you like to insert the array members? For select yourself, press - 1  For random selection press - 2 2 The original array: [627,732,706,935,471,362,841,187,306,663,636,982,319,35,5]  Method 1: The array of 100 minimal organs: [0,0,7,8,10,16,22,22,23,27,30,35,35,42,54,57,66,95,96,98,]  The number of comparisons made by method 1 is: 1747  Method 2: The array of 100 minimal organs: [0,0,7,8,10,16,22,22,23,27,30,35,35,42,54,57,66,95,96,98,]  The number of comparisons made by method 2 is: 2250</pre>	<pre>Mmn14.main({ }); Select the array size 200 Choose how few minimal organs you want to print 50 In what way would you like to insert the array members? For select yourself, press - 1  For random selection press - 2 2 The original array: [293,105,326,381,293,219,912,600,588,82,775,861,699,34]  Method 1: The array of 50 minimal organs: [4,4,5,24,24,31,35,47,52,60,75,79,82,82,94,95,95,98,98]  The number of comparisons made by method 1 is: 1086  Method 2: The array of 50 minimal organs: [4,4,5,24,24,31,35,47,52,60,75,79,82,82,94,95,95,98,98]  The number of comparisons made by method 2 is: 1159</pre>	<pre>Mmn14.main({ }); Select the array size 200 Choose how few minimal organs you want to print 8 In what way would you like to insert the array members? For select yourself, press - 1  For random selection press - 2 2 The original array: [51,28,919,96,73,890,251,857,322,198,156,562,641,284,29]  Method 1: The array of 8 minimal organs: [19,23,28,30,51,61,67,73]  The number of comparisons made by method 1 is: 500  Method 2: The array of 8 minimal organs: [19,23,28,30,51,61,67,73]  The number of comparisons made by method 2 is: 538</pre>	n= 200
<pre>Mmn14.main({ }); Select the array size 500 Choose how few minimal organs you want to print 100 In what way would you like to insert the array members? For select yourself, press - 1  For random selection press - 2 2 The original array: [766,34,291,367,374,908,157,138,855,538,305,344,297,387,6]  Method 1: The array of 100 minimal organs: [4,6,8,10,14,14,17,18,19,19,19,21,23,24,25,28,33,34,35,37]  The number of comparisons made by method 1 is: 2665  Method 2: The array of 100 minimal organs: [4,6,8,10,14,14,17,18,19,19,19,21,23,24,25,28,33,34,35,37]  The number of comparisons made by method 2 is: 3726</pre>	<pre>Mmn14.main({ }); Select the array size 500 Choose how few minimal organs you want to print 50 In what way would you like to insert the array members? For select yourself, press - 1  For random selection press - 2 2 The original array: [802,717,507,608,215,158,382,689,166,760,679,917,324,813]  Method 1: The array of 50 minimal organs: [3,7,11,13,14,15,19,19,19,26,26,27,27,28,35,40,40,47,48,]  The number of comparisons made by method 1 is: 1945  Method 2: The array of 50 minimal organs: [3,7,11,13,14,15,19,19,19,26,26,27,27,28,35,40,40,47,48,]  The number of comparisons made by method 2 is: 1458</pre>	<pre>Mmn14.main({ }); Select the array size 500 Choose how few minimal organs you want to print 8 In what way would you like to insert the array members? For select yourself, press - 1  For random selection press - 2 2 The original array: [395,222,407,775,725,922,6,565,342,718,293,880,527,888,366]  Method 1: The array of 8 minimal organs: [1,1,1,1,3,3,4,4]  The number of comparisons made by method 1 is: 1258  Method 2: The array of 8 minimal organs: [1,1,1,1,3,3,4,4]  The number of comparisons made by method 2 is: 1618</pre>	n= 500
<pre>Mmn14.main({ }); Select the array size 1000 Choose how few minimal organs you want to print 100 In what way would you like to insert the array members? For select yourself, press - 1  For random selection press - 2 2 The original array: [111,803,505,675,342,240,383,415,961,499,829,10,645,467,4,1]  Method 1: The array of 100 minimal organs: [0,1,1,3,4,4,6,6,7,9,10,10,10,11,14,14,15,16,17,17,18,19,1]  The number of comparisons made by method 1 is: 4023  Method 2: The array of 100 minimal organs: [0,1,1,3,4,4,6,6,7,9,10,10,10,11,14,14,15,16,17,17,18,19,1]  The number of comparisons made by method 2 is: 5497</pre>	<pre>Mmn14.main({ }); Select the array size 1000 Choose how few minimal organs you want to print 50 In what way would you like to insert the array members? For select yourself, press - 1  For random selection press - 2 2 The original array: [374,472,599,472,470,123,490,517,617,551,85,384,761,863,406]  Method 1: The array of 50 minimal organs: [1,3,4,4,5,7,9,10,12,13,13,14,15,15,16,16,18,18,19,20,22,22]  The number of comparisons made by method 1 is: 3071  Method 2: The array of 50 minimal organs: [1,3,4,4,5,7,9,10,12,13,13,14,15,15,16,16,18,18,19,20,22,22]  The number of comparisons made by method 2 is: 4233</pre>	<pre>Mmn14.main({ }); Select the array size 1000 Choose how few minimal organs you want to print 8 In what way would you like to insert the array members? For select yourself, press - 1  For random selection press - 2 2 The original array: [275,241,515,91,202,321,804,741,9,661,236,285,292,691,668,2]  Method 1: The array of 8 minimal organs: [2,2,2,5,5,6,6,8]  The number of comparisons made by method 1 is: 2394  Method 2: The array of 8 minimal organs: [2,2,2,5,5,6,6,8]  The number of comparisons made by method 2 is: 1894</pre>	n=1000

## סיכום

### שאלה 1

שיטה 1:

במקרה הגרוע:  $O(n \log n) = n \log(n) + 1 + k(\log(n) + \log(n-k))$   
במקרה הטוב:  $O(n) = n + 1 + k(\log(n) + \log(n-k))$

שיטה 2:

במקרה הטוב:  $O(n) = n + k \log k$   
במקרה הגרוע:  $O(n^2) = k \log k + \frac{n(1+n)}{2}$   
במקרה הממוצע:  $O(n \log n) = k \log k + n \log n$

### שאלה 2

#### מדוע עבור כל הרצה נקבל מספר השוואות שונה?

עבור שתי השיטות ניתן לראות כי הניתוח התיאורטי מתיישב עם ההרצות בכך שכמות ההשוואות שקיבלנו בפועל היא בקירוב "המקרה הממוצע", כלומר, כמות ההשוואות שקיבלנו היא גדולה מכמות ההשוואות של המקרה הטוב וקטנה מכמות ההשוואות של המקרה הגרוע. כלומר, לא התבצעה חריגה לכמות ההשוואות שצפינו לקבל.  
ובכל זאת נסביר את השוני בין מספר ההשוואות בין ההרצות:  
0

שיטה 1

טענה: ככל שהמערך המקורי יהיה יותר ממוין יתבצעו פחות השוואות.  
נימוק: אחרי כל הכנסה של איבר לתחתית הערמה מתבצע heapify שבמידת הצורך "מציף" את האיבר לכיוון ראש הרשימה, והעלאת האיבר בערמה כרוכה בהשוואות – אם האיבר שמעל קטן מימנו, וכל עוד התשובה היא שלילית מתבצעות עוד השוואות, אז ככל שהמערך יותר ממוין, אחרי הכנסה של כל איבר לתחתית הערימה יהיו מעליו פחות איברים שגדולים מימנו, וזה אומר שנסתפק בפחות השוואות כדי שהאיבר החדש יהיה במקומו.  
כמובן שהטענה נכונה גם בכיוון השני – ככל שהמערך פחות ממוין/יותר בסדר יורד, השיטה heapify תצטרך להעלות כל איבר יותר רמות למעלה מה שאומר – יותר השוואות. ולכן כמות ההשוואות משתנה בין המערכים השונים.

שיטה 2

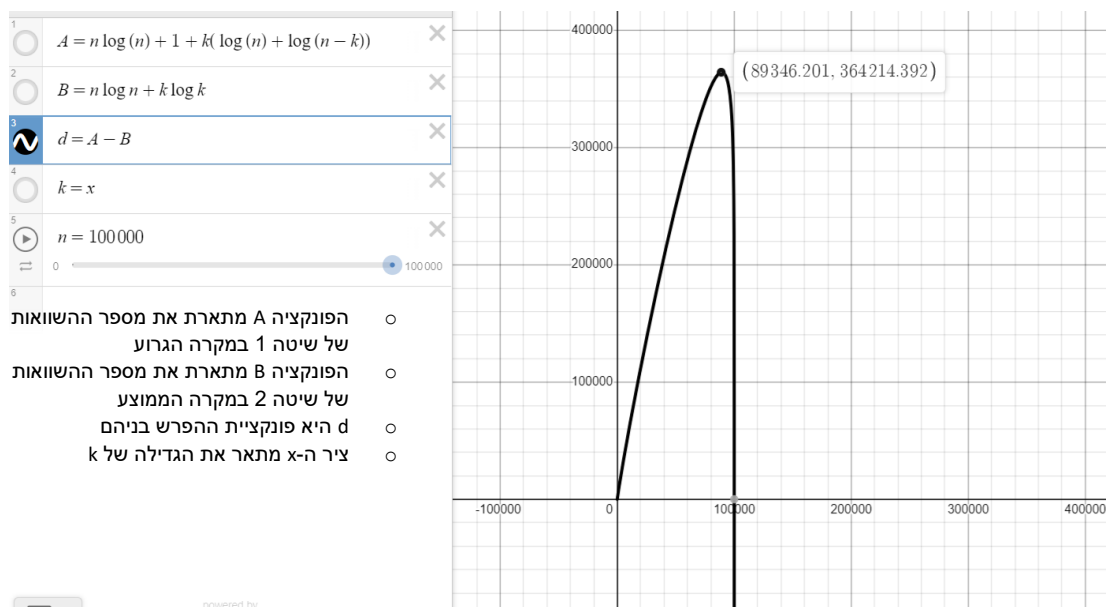
בשיטה זו השוני בין מספר ההשוואות תלויה בעיקר בבחירה הרנדומלית של השיטה Randomize-Select.  
בכל קריאה רקורסיבית לשיטה, מתבצעות  $n$  השוואות, וככל ש"התמזל מזלנו" והאינדקס שבו נמצא האיבר ה- $k$  בגודלו נבחר בבחירה הרנדומלית אחרי מספר נמוך יותר של קריאות, יתבצעו פחות השוואות, וכמובן שהטענה נכונה גם בכיוון השני.  
כמובן שסוג המערך גם יכול להשפיע – למשל אם בבחירה רנדומלית של הרצה כלשהי האינדקס שנבחר הוא  $i$  אז עבור מערך  $A$  שבו באינדקס  $i$  נמצא האיבר ה- $k$  בגודלו, החיפוש יסתיים, אך עבור מערך  $B$  שבו באינדקס  $i$  לא נמצא האיבר החיפוש ימשיך ואיתו יגדל מספר ההשוואות.

עוד גורם שאולי יכול להשפיע הוא גודל ה- $k$ . ככל שה- $k$  קרוב יותר לקצוות, יש יותר סיכוי שהשיטה Randomize-Partition תידרש לרוץ פחות פעמים, כי אחרי כל קריאה ל-Randomize-Partition טווח הבחירה של הקריאה הבאה מצטמצם, ומכאן השוני בין מספר ההשוואות בכל הרצה.

## השערה:

- בהתייחס למקרה הגרוע של שיטה 1 לעומת המקרה הממוצע של שיטה 2, כאשר ה-k שואף ל- $\frac{9n}{10}$  כך השיטה הראשונה יעילה יותר מבחינת כמות השוואות.

דוגמה ע"י גרף:



ניתן לראות שאכן  $k$  שואף ל- $\frac{9n}{10}$  הפונקציה קרובה לפיק שלה – הפרש ההשוואות הוא המקסימלי

דוגמאות ר'צה:

$$K = \frac{n}{10,000} \text{ ומספר ההשוואות בשיטה 2 גדול פי 2.14}$$

$$K = \frac{9n}{10} \text{ ומספר ההשוואות בשיטה 2 גדול פי } 11.96$$

```
Mmn14.main({ });
Select the array size
100000
Choose how few minimal organs you want to print
10
In what way would you like to insert the array members?
For select yourself, press - 1

For random selection press - 2
2
The original array:
[380,49,81,700,581,240,406,165,526,908,466,825,35,96,709,7

Method 1:

The array of 10 minimal organs:
[0,0,0,0,0,0,0,0,0,0]

The number of comparisons made by method 1 is: 228255

Method 2:
The array of 10 minimal organs:
[0,0,0,0,0,0,0,0,0,0]

The number of comparisons made by method 2 is: 487659
```

[illegible]

