**ממ"ן 13- לירן טרכטר(206606782) ועדי שמש(206015083)**

אלגוריתם מיון מהיר 1- נרשם כמתואר בספר בקוד נקרא quickSort1 ומשתמש ב- partition1

אלגוריתם מיון מהיר 2- עובד באופן דומה כמו מהיון מהיר 1 אך משתמש בpartition2

Partition2- השינוי בין partition1 ל-partition2 הוא בחירת איבר הציר- partition2 בוחר את איבר הציר כך שבין 3 האיברים האחרונים הוא לוקח את האיבר עם הערך האמצעי בעזרת הפונקציה MiddleOfThree. (כלומר 1,5,8 ייקח את המספר 5).

ביצוע ההשוואות בשני האלגוריתמים נמצא ב- partition בתוך הלולאה כמשתנה count

נכונות האלגורתמים:

quickSort1 דיון נכונות בספר

quickSort2 אם המערך בגודל הקטן מ3 עובד באופן זהה לquickSort1, אחרת ההבדל הוא רק בבחירת איבר הציר ועובד באופן זהה.

סיבוכיות:

quickSort1- במקרה הגרוע O(n2)

במקרה הממוצע O(nlog(n))

quickSort2- במקרה הגרוע O(n2) אמנם לא נגיע לn2 אך כן נוכל להגיע למצב בו החלוקות יהיו באופן לא שוויוני למשל תת מערך של שני איברים ותת מערך של n-2 (איבר הציר ייבחר תמיד כאיבר המקסימלי פחות אחד)

במקרה הממוצע O(nlog(n))

השארה- quickSort2 ימיין באופן מהיר יותר, אך עדיין בגבולות nlogn למקרה הממוצע.

מקרה טוב-

מקרה טוב לשני האלגוריתמים יגרמם בכך שהמערך יתחלק באיטרציה הראשונה לתת מערך בגודל זהה ככל האפשר . לכן המערך יראה באופן הבא : {\_,\_,\_....,n-1/2,n/2}

מקרה גרוע-

מקרה גרוע לשני האלגוריתמים יגרמם כך שהמערך יתחלק באיטרציה הראשונה לתת מערך בגודל 1 ותת מערל בגודל n-1 . לכן המערך יראה באופן הבא : {\_,\_,\_....n-1,n}

מקרה ממוצע-

חישוב כל ההשוואות חלקי מספר הפרמוטציות

טבלת השוואות-

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n | מספר השוואות ממוצע quickSort1 | מספר השוואות ממוצע quickSort2 | a | מספר ההשוואות ממוצע עבור a המקרים הגרועים quickSort1 | מספר ההשוואות ממוצע עבור a המקרים הגרועים quickSort2 | מספר ההשוואות ממוצע עבור a המקרים הטובים quickSort1 | מספר ההשוואות ממוצע עבור a המקרים הטובים quickSort2 |
| 7 | 13.485 | 12.073 | 0.023 | 18.4 | 16.3 | 11.66 | 11.66 |
| 8 | 16.921 | 15.124 | 0.017 | 23.3 | 20.304 | 14.833 | 14.291 |
| 9 | 20.579 | 18.375 | 0.013 | 28.485 | 24.536 | 18.5 | 17.458 |
| 10 | 24.437 | 21.806 | 0.011 | 33.921 | 28.975 | 22.066 | 20.825 |
| 11 | 28.477 | 25.402 | 0.009 | 39.579 | 33.598 | 26.233 | 24.2 |
| 12 | 32.683 | 29.149 | 0.007 | 45.437 | 38.389 | 30.133 | 27.79 |

ניתוח טבלת השוואות-

חישוב a -

בטבלה מתואר ממוצע מספר ההשוואות עבור המקרה הטוב, הרע והממוצע. בכל אחד מהמקרים חושב סכום ההשוואות חלקי סך המקרים. ניתן לראות כי בכל המקרים שנבדקו quickSort2 יעילה יותר אך לא באופן משמעותי אסימפטוטי. (נשים לב כי עבור n קטן שווה מ7 במקרה הטוב הממוצע השוואות שווה). כמו שחשבנו quickSort2 ממיין באופן יעיל יותר אך בגבולות nlogn.

מספר השוואות מצופה-

n=7:

בפועל קבלנו עבור quickSort1 67968 השוואות

ועבור quickSort2 קבלנו 60851 השוואות

n=8:

בפועל קבלנו עבור quickSort1 682272 השוואות

ועבור quickSort2 קבלנו 609807 השוואות

n=9:

בפועל קבלנו עבור quickSort1 7467840 השוואות

ועבור quickSort2 קבלנו 6668027 השוואות

n=10:

בפועל קבלנו עבור quickSort1 88678080 השוואות

ועבור quickSort2 קבלנו 791324450 השוואות

n=11:

בפועל קבלנו עבור quickSort1 1136712960 השוואות

ועבור quickSort2 קבלנו 1013987599 השוואות

n=12:

בפועל קבלנו עבור quickSort1 15655438080 השוואות

ועבור quickSort2 קבלנו 13962485527 השוואות

דוגמאות הרצה

תמונה שמכילה טקסט, עיתון, צילום מסך, קבלה

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, עיתון, קבלה, צילום מסך

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, מקורה, צילום מסך, מסמך

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי