Quick Sort

~ עמית מאיר שובל

מה זה?

אלגוריתם המשמש לסידור מערך בצורת "רגל ציר".

המטרה בשיטת סידור זו היא קביעה של נקודה על המערך ושכל האיברים מצד אחד יהיו קטנים מה-ציר, וכל המספרים המצד השני יהיו גדולים מה-ציר. הצדדים של הציר נקראים "מחיצות - Partitions".

דוגמה עבור סידור זה:

* מחוג השעה, ברגע שהמחוג מצביע על שעה מסוימת כל שער המספר מצד ימין של השעון יהיו *קטנים* מהשעה, וכל המספרים משמאל יהיו *גדולים*.

בשיטת סידור זו אנחנו מבצעים באופן ישיר סידור לפי גודל. כך שבסיום מחזור הסידור המערך יהיה ממוין לפי מה שיותר גדול או קטן מנקודת ציר, אבל לא בהכרח בסדר עולה או יורד.

(נוכל להשתמש בשיטת סידור זו כדי לסדר מערך בסדר עולה או יורד מתחילתו ועד סופו בעזרת רקורסיה)

בהינתן המערך הבא:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 5 | 3 | 6 | 8 | 1 | 2 | 7 |

נבחר באלמנט האחרון כנקודת הציר (קריאה משמאל לימין).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 5 | 3 | 6 | 8 | 1 | 2 | 7 |

נרצה לגרום שכל המספרים מצד אחד (שמאל) ערכם יהיו נמוכים מהערך של נקודת הציר (4).  
לצד השני נרצה להעביר את כל המספרים שערכם גדול מנקודת הציר.

* כדי לעבור על איברי המערך נבצע זאת עם שני משתנים:  
  i = למיקום של האלמנט הראשון פחות 1.  
  j = למיקום האלמנט השני והוא ירוץ עד גודל המערך פחות 1.
* נבצע השוואה בין הציר ובין אלמנט j.
  + אם j גדול או שווה לציר נעבור למספר הבא.
  + אם j קטן מהציר:
    - נגדיל את הערך של i ב-1
    - נחליף בין המקומות של ה-i וה-j.
    - נגדיל את ערך j ב-1.

בסופה של הריצה המקום של הציר אמור להיות המיקום הבא שאחרי i, כלומר המיקום של הציר יהיה i+1.

ועכשיו מצאנו את המיקום שלו, אז נחליף אותו למיקום הזה.

כך ירצנו מחיצה שמצד אחד יש מספרים שהם גדולים מה-ציר ומחיצה בה המספרים קטנים מהציר.

j = 7, i = 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 5 | 3 | 6 | 8 | 1 | 2 | 7 |

j(7) > 4

--------------------------------------------------------------

j = 2, i = 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 5 | 3 | 6 | 8 | 1 | **2** | **7** |

j(2) < 4

i = i+1

arr[j] מחליף עם arr[i]

--------------------------------------------------------------

j = 1, i = 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 5 | 3 | 6 | 8 | 1 | 7 | 2 |

j(1) < 4

i = i+1

arr[j] מחליף עם arr[i]

--------------------------------------------------------------

j = 8, i = 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 5 | 3 | 6 | 8 | **7** | **1** | 2 |

j(8) > 4

--------------------------------------------------------------

j = 6, i = 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 5 | 3 | 6 | 8 | 7 | 1 | 2 |

j(6) > 4

--------------------------------------------------------------

j(3), i = 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 5 | 3 | 6 | 8 | 7 | 1 | 2 |

j(3) < 4

i = i+1

arr[j] מחליף עם arr[i]

--------------------------------------------------------------

j = 5, i = 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 5 | **7** | 6 | 8 | **3** | 1 | 2 |

j(5) > 4

--------------------------------------------------------------

החלפת הציר

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **8** | 5 | **7** | 6 | **4** | 3 | 1 | 2 |

i = 3

המיקום של נקודת הציר במערך הוא ערך i+1

לאחר מציאת נקודת האמצע של המערך נחלק אותו ל-2 מערכים לפי הצדדים ונסדר אותם גם כן.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **8** | 5 | **7** | 6 | **4** | 3 | 1 | 2 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 5 | **7** | 6 |  | 3 | 1 | 2 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 5 | **7** | 6 |  | 3 | 1 | 2 |

קביעת נקודת ציר

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 5 | **7** | 6 |  | 3 | 1 | 2 |

תחילת השוואה

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 5 | **7** | 6 |  | 3 | 1 | 2 |

מחליפים

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 5 | 7 | 6 |  | 3 | 1 | 2 |
| 8 | 5 | 6 | 7 |  | 3 | 2 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 5 | 6 | 7 |  | 3 | 2 | 1 |

בסיום שלב זה 3 כבר ממוקם נכון במקומו

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 6 | 5 | 7 |  | 3 | 2 | 1 |
| 8 | 6 | 5 | 7 |  | 3 | 2 | 1 |

בסיום שלב זה 8 ממוקם נכון

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 6 | 5 | 7 |  | 3 | 2 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 6 | 5 | 7 |  |  | 2 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 6 | 5 | 7 |  |  | 2 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 6 | 5 | 7 |  |  | 2 | 1 |

בסיום שלב זה ה-2 ממוקם נכון

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 6 | 5 | 7 |  |  | 2 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 6 | 7 | 5 |  |  |  | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 6 | 7 | 5 |  |  |  | 1 |

בסיום שלב זה 1 ממוקם נכון

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 7 | 6 | 5 |  |  |  | 1 |

בסיום שלב זה 6 מוקם נכון

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 7 |  | 5 |  |  |  |  |

הגענו לקצה הרקורסיה ובנקודה זו המערך מסודר

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** |

Code

#include <stdio.h>

//החלפה בין שני האלמנטים

void swap(int \*a, int \*b) {

    int temp = \*a;

    \*a = \*b;

    \*b = temp;

}

//מחלק את המערך ומחזיר את המיקום במערך של נקודת הציר

int partition(int arr[], int low, int high) {

    int pivot = arr[high];

    int i = low - 1;

    for (int j = low; j <= high - 1; j++) {

        if (arr[j] <= pivot) {

            i++;

            swap(&arr[i], &arr[j]);

        }

    }

    swap(&arr[i + 1], &arr[high]);

    return i + 1;

}

//פונקציה רקורסיבית המבצעת את הסידור

void quickSort(int arr[], int low, int high) {

    if (low < high) {

        //מחלק את המערך ומקבל את נקודת הציר

        int pivotIndex = partition(arr, low, high);

        //באופן רקורסיבי מחלק את תת המערך

        quickSort(arr, low, pivotIndex - 1);

        quickSort(arr, pivotIndex + 1, high);

    }

}

int main() {

    int arr[] = {7, 2, 1, 8, 6, 3, 5, 4};

    size\_t numElements = sizeof(arr) / sizeof(arr[0]);

    printf("Original array: ");

    for (size\_t i = 0; i < numElements; i++) {

        printf("%d ", arr[i]);

    }

    quickSort(arr, 0, numElements - 1);

    printf("\nSorted array: ");

    for (size\_t i = 0; i < numElements; i++) {

        printf("%d ", arr[i]);

    }

    return 0;

}