

מציאת מינימום ומקסימום במערך

תיאור הבעיה

נתון מערך A בעל n איברים - יש למצוא את הערך המינימלי והמקסימלי במערך זה.

5	7	8	4	9	14	3	52	16	2
max								min	

פתרון הבעיה

עבור בעיה זו ישנם 4 אפשרויות פתרון, כל אפשרות מציגה סיבוכיות השוואה שונה, המטרה היא להשתמש בכמות השוואות קטנה ככל הניתן כדי לייעל את סיבוכיות האלגוריתם.

❖ אפשרות א':

⇐ נפעיל את האלגוריתם min שהצגנו למציאת מינימום.

⇐ נפעיל את האלגוריתם max שהצגנו למציאת מקסימום.

⇐ נחזיר את התוצאות.

```
public static int getMax(int[] arr) {
    int ans = arr[0];
    for(int i = 1 ; i < arr.length ; i++) {
        if(ans < arr[i])
            ans = arr[i];
    }
    return ans;
}

public static int getMin(int[] arr) {
    int ans = arr[0];
    for(int i = 1 ; i < arr.length ; i++) {
        if( ans > arr[i])
            ans = arr[i];
    }
    return ans;
}

public static void main(String[] args) {
    int[] arr = {84,31,3,1,567,4,2,93202,32,3};
    System.out.println(getMax(arr));
    System.out.println(getMin(arr));
}
```

~ סיבוכיות השוואה: סה"כ נקבל $2n - 2$ השוואות.

❖ אפשרות ב':

⇐ נמזג את 2 האלגוריתמים $\min(A)$, $\max(A)$ לאלגוריתם אחד.

```

public static int[] minMax(int[] arr) {
    int max = arr[0];
    int min = arr[0];

    for(int i = 1 ; i < arr.length; i++) {
        if(max < arr[i])
            max = arr[i];
        if(min > arr[i])
            min = arr[i];
    }

    return new int[] { max, min };
}

public static void main(String[] args) {
    int[] arr = {84,31,3,1,567,4,2,93202,32,3};
    System.out.println(Arrays.toString(minMax(arr)));
}

```

~ סיבוכיות השוואה: סה"כ נקבל $2n - 2$ השוואות.

❖ אפשרות ג':

⇐ נצמצם עוד השוואה בעזרת קביעת min,max בהתחלה.

```

public static int[] improvedMinMax(int[] arr) {
    int max = 0;
    int min = 0;

    if(arr[0] > arr[1]) {
        max = arr[0];
        min = arr[1];
    } else{
        max = arr[1];
        min = arr[0];
    }

    for( int i = 2 ; i < arr.length ; i++ ) {
        if(max < arr[i]) {
            max = arr[i];
        }
        if(min > arr[i]){
            min = arr[i];
        }
    }
}

```

```

    return new int[] {max, min};
}

public static void main(String[] args) {
    int[] arr = {84,31,3,1,567,4,2,93202,32,3};
    System.out.println(Arrays.toString(improvedMinMax(arr)));
}

```

~ סיבוכיות השוואה: סה"כ נקבל $2(n-2) + 1 = 2n - 4 + 1 = 2n - 3$.

❖ אפשרות ד':

⇐ ננסה לצמצם עוד את מספר ההשוואות - נעבור על המערך בזוגות.

```

public static int[] finalSolution(int[] arr){
    int max = 0;
    int min = 0;

    if(arr[0] > arr[1]) { // 1 comparison
        max = arr[0];
        min = arr[1];
    } else {
        max = arr[1];
        min = arr[0];
    }

    // n-2 values, 2*jumps which is n-2/2, in each iteration we make 3 comparisons
    // in conclusion, n-2/2 + n-2/2 + n-2/2 = 3(n-2)/2
    for(int i = 2 ; i < arr.length - 1 ; i+=2) {

        if(arr[i] < arr[i+1]) { // 3 comparisons

            if(min > arr[i]) {
                min = arr[i];
            }
            if(arr[i+1] > max){
                max = arr[i+1];
            }

        } else { // 3 comparisons
            // if arr[i] >= arr[i+1]
            if(min > arr[i+1]) {
                min = arr[i+1];
            }
            if(max < arr[i]) {
                max = arr[i];
            }
        }
    }
}

```

```

    }

}

if(arr.length % 2 != 0) { // 2 comparisons
    if(arr[arr.length-1] > max) {
        max = arr[arr.length-1];
    }
    else if(arr[arr.length-1] < min) {
        min = arr[arr.length-1];
    }
}
// in conclusion  $1 + 3n/2 + 2 = 3n/2 + 4 = O(3n/2)$ 
return new int[] { max, min};
}

public static void main(String[] args) {
    int[] arr = {84,31,3,1,567,4,2,93202,32,3};
    System.out.println(Arrays.toString(finalSolution(arr)));
}

```

~ סיבוכיות השוואה: סה"כ נקבל $\frac{3n}{2} - 3 + 3 = \frac{3n}{2}$ $\frac{3n-6}{2} + 3 = \frac{3n-2}{2} + 1 + 2 = \frac{3n-2}{2}$ השוואות.