שאלות

שאלה 1

כתוב פונקציה סטטית שמקבלת מערך של מספרים שלמים וממיינת אותו כך שכל המספרים הזוגים נמצאים בתחילת המערך, והמספרים האי- זוגיים נמצאים בסוף המערך. סיבוכיות נדרשת $O(n)$.

דוגמא: קלט: $\{-11, 6, -8, 14, 12, 1, 27, 30, 13, 5, 246, 1, 2\}$

פלט: $[2,\ 6,\ -8,\ 14,\ 12,\ 246,\ 30,\ 27,\ 13,\ 5,\ 1,\ 1,\ -11]$

שאלה 2

כתוב פונקציה סטטית שמקבלת מערך ממוין בסדר עולה של מספרים שלמים.

הפונקציה מחזירה true אם במערך יש שני איברים שסכומם שווה לאפס, אחרת היא מחזירה false סיבוכיות נדרשת $O(n)$.

שאלה 3

כתוב פונקציה שמקבלת מערך של מספרים שלמים ומדפיסה את שני האברים שערך מוחלט של ההפרש בינם הוא גדול היותר. מהי סיבוכיות של האלגוריתם?

דוגמא: קלט: $int\ []arr\ =\ \{1, 4, 9, 17, 23, -1, 14\}$

פלט: $a1\ =\ -1,\ a2\ =\ 23$

שאלה 4

משחק – ניחוש. המשתמש חושב על מספר בין 1 ל-1000. כתוב תוכנית המנחשת את המספר של המשתמש **יעילה ככל האפשר** .

התוכנית מדפיסה מספר בין 0 ל-1000 ומבקשת את המשתמש לבחור באפשרות המתאימה:

האם זה המספר שחשבת אליו?

האם המספר שחשבת אליו קטן ממספר שהדפסתי?

האם המספר שחשבת אליו גדול ממספר שהדפסתי?

התשובה של המשתמש היא: <1 או 2 או 3>

המשתמש בוחר באפשרות המתאימה ועונה לתוכנית. המשחק נמשך עד שהמשתמש מקבל את המספר שלו. המשתמש בוחרת

במידה והמספר זהה למספר של המשתמש (המשתמש מקליד 1) התוכנית מדפיסה: "WIN!" והמשחק מסתיים. בסוף המשחק התוכנית מדפיסה את מספר השלבים שהתבצעו עד ניחוש המספר.

שאלה 5

נתון מערך שמכיל מספרים שלמים מ-1 עד 100.צריך למיין את המערך בסיבוכיות של $O(N)$.

דוגמא: קלט: $int\ []arr = \{98,2,3,1,0,0,0,3,98,98,2,2,2,0,0,0,2\};$

פלט: $0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 2,\ 2,\ 2,\ 2,\ 2,\ 3,\ 3,\ 98,\ 98,\ 98$.

שאלה 6

ממשו את *Binary Search* (בונוס בשביל עצמכם , נסו לממש אותו גם בצורה איטרטיבית וגם בצורה ריקורסיבית).

שאלה 7

ממשו את *Merge Sort*.

שאלה 8

ממשו את *Quick Sort*.

שאלה 9 בונוס

כתבו פונקציה שבהינתן מערך של מספרים ממוינים ומספר, צריכה להחזיר את מיקום המספר במערך אם המספר קיים במערך ואם לא, את מיקומו אילולא היה קיים במערך.

לדוג- $nums = [1,3,5,6],\ target = 5$ הפלט יהיה 2.

$nums = [1,3,5,6],\ target = 2$ הפלט יהיה 1.

חתימה הפונקציה : $public\ static\ int\ searchInsert(int[]\ nums,\ int\ target)$

רמז - חיפוש בינארי איטרטיבי

פתרונות

פתרון שאלה 1

```
public static void q1(int arr[]){
        int low = 0;
        int high = arr.length -1;
        while(low<=high){
            if(arr[low]%2==0)//number is even and is in low position.
                low++;
            else if(arr[high]%2==1)//number is odd and in a high
position
                high--;
            else{//we found a need to swap between odd number in low
position and even number in high position.
                int temp = arr[low];
                arr[low] = arr[high];
                arr[high] = temp;
                high--;
                low++;
            }
        }
    }
```

פתרון שאלה 2

```
public static boolean sum0(int[]a){
        int start =0 ;
        int end = a.length-1;
        while(start<end){
            if(a[start]+a[end]==0)
                return true;
            else if(a[start]+a[end]<0)
                start++;
            else
                end--;
        }
        return false;
    }
```

פתרון שאלה 3

```
//complexity O(n)
public static  void absmax(int[]a ){
    int min= Integer.MAX_VALUE;
    int max= Integer.MIN_VALUE;
    for (int i = 0; i < a.length; i++) {
        if(min>a[i])
            min = a[i];
    }
    for (int i = 0; i < a.length; i++) {
        if(max<a[i])
            max= a[i];
    }
    System.out.println("a1 = " + min + "  a2 = " + max);
}
```

פתרון שאלה 4

```
   public static void game() {
        int a = 0;
        int low = 0;
        int high = 1000;
        int middle = (low + high) / 2;
        while (low < high) {
            System.out.println(middle);
            Scanner sc = new Scanner(System.in);
            System.out.println("Is this your number? if so enter 1");
            System.out.println("If your number is smaller then the
number on the screen please enter 2");
            System.out.println("If your number is greater then the
number on the screen please enter 3");
            a = sc.nextInt();
            if (a == 1) {
                System.out.println("Win!!");
                sc.close();
                return;
            } else if (a == 2) {
                high = middle - 1;
                middle = (low + high) / 2;
            } else {
                low = middle + 1;
                middle = (low + high) / 2;
            }
        }
    }
```

פתרון שאלה 5

```
//complexity O(n)
public static int[] hundredsort(int[] a){
    int counter =0;
    int[] a2 = new int[a.length];
    for (int i = 0; i <=100 ; i++) {
        for (int j = 0; j < a.length; j++) {
            if(a[j]==i)
                a2[counter++]=i;
        }
    }
    return a2;
}
```

פתרונות לשאלות 6,7,8 נמצאות במצגות של התרגול.

פתרון שאלה 9

```
public static int searchInsert(int[] nums, int target) {
  if(nums == null || nums.length == 0) return 0;//edge cases
  int n = nums.length;
  int l = 0;
  int r = n - 1;
  while(l < r){
    int m = l + (r - l)/2;//same as (l+r)/2 but avoiding overflow

    if(nums[m] == target) return m;
    else if(nums[m] > target) r = m; // right could be the result
    else l = m + 1; // m + 1 could be the result
  }

  // 1 element left at the end
      if (nums[l]<target){
          return l+1;
      }
      else{
          return l;
      }
```

```
}
```