

## פתרון מוצע לבחינת מה"ט באלגוריתמיקה ותכנות ב JAVA

מועד א' תשפ"ב, מרץ 2202  
 מחבר: זהבה לביא, מכללת אורט רחובות

חלק א'

### שאלה מס. 1

התוכנית קולטת בכל סיבוב לולאה צמד ערכים ובודקת שהם אינם ערכים נגדיים. אם אינם נגדיים, מספר הזוגות המחושב גדל ב-1, מתבצעת בדיקה אם הערכים הנקלטים זהים וכן אם כל אחד מהם חיובי.

```

        public static void t1() {
            int num1, num2;
            int count=0, posAmount=0, equalCount=0;
            num1=in.nextInt();
            num2=in.nextInt();
            while(num2!=num1*-1&& num2*-1!=num1) {
                count++;
                if (num2==num1)
                    equalCount++;
                if (num2>0)
                    posAmount+=num2;
                if (num1>0)
                    posAmount+=num1;
                num1=in.nextInt();
                num2=in.nextInt();
            }
            System.out.println("count="+count);
            System.out.println("positive amount="+posAmount);
            System.out.println("equals="+equalCount);
        }
    
```

שאלה מס. 2

הפעולה מגרילה מספר אקראי דו ספרתי חיובי. אם ערך ספרת האחדות אינו גדול מספרת העשרות, מוגרל מספר חדש לאותו המקום במעריך.

```
public static int [] t2(int num) {  
    int [] arr = new int [num];  
  
    for(int i=0;i<num; i++) {  
        arr[i]=(int)(Math.random() * (90) + 10);  
        if (arr[i]%10 <=arr[i]/10)  
            i--;  
    }  
    return arr;  
}
```

### שאלה מס. 3

א. הפעולה בודקת אם גודל המערך מתחלק ב-3. אם לא, מוחזר שקר. אחרת, נבדק כל איבר מהשליש הראשון עם איבר מהשליש השני ומהשליש השלישי.

```
public static boolean t3(int []a) {
    if (a.length%3!=0) {
        return false;
    }
    int n=a.length;
    for(int i=0;i<a.length/3; i++) {
        if (!(a[i]==a[i+n/3]&&a[i]==a[i+n*2/3]))
            return false;
    }
    return true;
}
```

ב. זמן הריצה יהיה  $O(n)$ , כלומר  $O(n)$ . בהנחה שמתבצעת סריקה של כל השליש הראשון של המערך.

### שאלה מס. 4

```
public static void t4() {
    int [] a = new int [6];
    int amount;
    int room= in.nextInt();
    while (room>=1 && room <=6) {
        amount=in.nextInt();
        if (a[room-1]+amount<200)
            a[room-1]+=amount;
        room= in.nextInt();
    }
}
```

```

        if (a[0]%8==0)
            System.out.println("Slices Pizza:"+ a[0]/8);
        else
            System.out.println("Slices Pizza:"+(a[0]/8+1));
        if (a[0]%20==0)
            System.out.println("tropical:"+ a[0]/20);
        else
            System.out.println("tropical:"+(a[0]/20+1));
        for (int i=1;i<a.length; i++) {
            if (a[i]<10) {
                System.out.println("room #"+(i+1)+" is empty");
            }
            if (a[i]<200) {
                System.out.println("empty places:"+(200-a[i])+"
                    in room "+(i+1));
            }
        }
    }
}

שאלה מס. 5
א.
    public String status() {
        return this.preformar+"/"+this.name+": "+this.length;
    }

```

ב. הנחת יסוד: מערך השירים מלא .

```
public boolean exist(String nSong, String pSong) {
    for (int i=0;i<this.songs.length; i++) {
        if
        (this.songs[i].getName().equals(nSong)&&this.songs[i].getPr
        eformar().equals(pSong)) {
            return true;
        }
    }
    return false;
}
```

ג.

```
public static String largeDiscName(Disc d1, Disc d2) {
    if (d1.songs.length>d2.songs.length)
        return d1.discName;
    else if (d1.songs.length<d2.songs.length)
        return d2.discName;
    else return "equals";
}
```

### שאלה מס. 6

א. נתון מערך  
 $arr = \{12, 13, 1, 10, 34\}$

ערך מוחזר	arr[i]	if3	if2	if1	z	y	x	i
					0	0	12	
	13			TRUE	0	12	13	1
	1	TRUE	FALSE	FALSE	1			2
	10	TRUE	FALSE	FALSE	10			3
	34			TRUE	12	13	34	4
59								

ב. הפעולה מחזירה את סכום 3 הערכים הגדולים ביותר במערך המתקבל.

ג. סיבוכיות הפעולה  $O(n)$ . מתבצעת סריקה של כל המערך.

חלק ב'

### שאלה מס. 7

א.

```

public boolean isImproved() {
    for(int i=0; i < this.arrGradesSemesterA.length; i++) {
        if (this.arrGradesSemesterA[i] != -
            1 && this.arrGradesSemesterB[i] != -
            1 && this.arrGradesSemesterA[i] > this.arrGradesSemesterB[i])
            return false;
    }
    return true;
}
    
```

ב.

```
public static String[] improvedStudents(Student [] arr) {
    int cnt=0;
    for(int i=0;i <arr.length; i++) {
        if (arr[i].isImproved())
            cnt++;
    }
    if (cnt>0){
        int index=0;
        String [] id = new String [cnt];
        for(int i=0;i <arr.length; i++) {
            if (arr[i].isImproved())
                id[index]arr[i].getId();
            index++;
        }
        return id;
    }
    return null;
}
```

### שאלה מס. 8

א.

```
public class Participant {
    private String name;
    private String id;
    private int price;
}

public class VickeryAuction {
    private String product;
    private int realPrice;
    private Participant[]arr;
    private int current;
}
```

```

        ב.8
public boolean addParticipant(Participant p) {
    if (this.current<this.arr.length&&
        p.getPrice()>0.5*this.realPrice){
        this.arr[this.current]=p;
        this.current++;
        return true;
    }
    return false;
}

        ג.8
    public Winner getWinner() {
        int max=this.arr[0].getPrice();
        int max1 = max;
        String name=this.arr[0].getName();
        for(int i=1;i <this.current; i++) {
            if (this.arr[i].getPrice()>max){
                max1=max;
            }
            max=this.arr[i].getPrice();
            name=this.arr[i].getName()
        }
        if (this.arr[i].getPrice()>max1){
            max1=this.arr[i].getPrice();
        }
    }
    return new Winner(name, max1);
}
    
```



## שאלה מס. 9

```
public static boolean isPretty(String str){
    int [] capitalLetters= new int [26];
    int [] smallLetters= new int [26];
    for (int i=0; i<str.length(); i++)
    {
        if (str.charAt(i) >='A' && str.charAt(i) <='Z' ){
            capitalLetters[str.charAt(i)-'A']++;
        }
        else if (str.charAt(i) >='a' && str.charAt(i) <='z' ){
            smallLetters[str.charAt(i)-'a']++;
        }
        else return false;
    }
}
```

## שאלה מס. 10

.א.

```
public static boolean isGold(int num1, int num2){
    if (num1%10 !=num2%10)
        return false;
    while(num1!=0 && num2!=0){
        num1 = num1/10;
        num2= num2/10;
    }
    return num1 == 0 && num2==0;
}
```

ב.

```
public static void goldCouples(int []arr){
    boolean found=false;
    for(int i=0; i<arr.length; i++)
        for(int j=i+1; j<arr.length; j++)
            if (isGold(arr[i], arr[j])){
                found=true;
                System.out.println("first:"+arr[i]+", "+ "second:"+arr[j]);
            }
    if (!found) System.out.println("no gold numbers");
}
```

ג.

סיבוכיות זמן הריצה של הפעולות:

סעיף א – בהנחה שאורך המספר הראשון המתקבל הוא באורך  $n$  ואורך המספר השני המתקבל הוא  $m$ , זמן הריצה יהיה  $O(\min(n,m))$ . פירוק מספרים לספרות. אם אורך המספרים זהה – זמן הריצה יהיה  $O(n)$ .

סעיף ב: בהנחה שאורך המערך המתקבל הוא  $z$ , כל איבר נבדק מול כל האיברים שאחריו ולכן  $O(z^2)$ . עבור כל זוג נבדקת הפעולה מסעיף א ולכן סה"כ זמן הריצה  $O(z^2) * O(\max(n,m))$ .

## שאלה מס. 11

א.

```
public static boolean specialNumber(int num){
    int sumOdd=0;
    int sumEven=0;
    while (num>0){
        if (cnt%2==0)
            sumEven+=num%10;
        else
            sumOdd+=num%10;
        cnt++;
        num=num/10;
    }
    return cnt>=2 && sumEven==sumOdd;}
}
```

```

        }
        public static boolean isSpecialArray(int [] arr){
            int i=0;
            while( i<arr.length && specialNumber(arr[i])
                i++;
            while( i<arr.length && !specialNumber(arr[i])
                i++;
            return i==arr.length;
        }
    }

    public static int [] buildSpecialArray(int [] arr1, int [] arr2){
        int [] temp1 = new int [arr1.length +arr2.length];
        int [] temp2 = new int [arr1.length +arr2.length];
        int index1=0, index2=0;
        for(int i=0;i<arr1.length; i++){
            if( specialNumber(arr1[i])
                {
                    temp1[index1]=arr1[i];
                    index1++;
                }
            else{
                temp2[index2]=arr1[i];
                index2++;
            }
        }
        for(int i=0;i<arr2.length; i++){
            if( specialNumber(arr2[i])
                {
                    temp1[index1]=arr2[i];
                    index1++;
                }
        }
    }

```

```

        else{
            temp2[index2]=arr2[i];
            index2++;
        }
    }
    for (int i=0;i<temp2.length && temp2[i]!=0; i++){
        temp1[index1]=temp2[i];
        index1++;
    }
    return temp1;
}

```

## שאלה מס. 12

### 1.א

```

public Flight(int code, String destination, int hour, int minute,
               int duration, int numOfPassengers){
    this.code=code;
    this.destination = destination;
    this.departureTime = new Time (hour, minute);
    this.duration = duration;
    this.numOfPassengers=numOfPassengers;
    if (this.numOfPassengers <MAX_CAPACITY)
        this.isFull=false;
    else
        this.isFull=true;
}

```

### 2.א

```
public boolean addPassengers (int num){
    if (this.numOfPassengers+num<=MAX_CAPACITY)
    {
        this.numOfPassengers+=num;
        if (this.numOfPassengers==MAX_CAPACITY)
            this.isFull=true;
        return true;
    }
    else return false;
}
```

### 3.א

```
public Time getArrivalTime(){
    Time t;
    int hour=this.departureTime.getHour();
    int minute = this.departureTime.getMinute();
    hour +=this.duration/60;
    minutes+=this.duration%60;
    t=new Time(hour+minutes/60, minutes%60);
    return t;
}
```

### 1.ב

```
public Time firstFlightToPlace(String place)
    Time t=null;
    for(int i=0; i<this.numOfFlights ; i++){
        if (this.flightsSchedule[i].getDestination().equals(place))
            if (t==null)
                t=this.flightsSchedule[i].getDepartureTime();
            else if (t.before(this.flightsSchedule[i].getDepartureTime()))
                t=this.flightsSchedule[i].getDepartureTime();
    }
    return t;}
```

## 2.ב.

```

public boolean MovePassengers(String codeF, int num)
    int i=0;
    while (i<this.numOfFlights &&
!this.flightsSchedule[i].getCodeF().equals( codeF))
        i++;
    if (i<this.numOfFlights){
for(int index=0; index<this.numOfFlights ; index++){
    if (index !=i &&
this.flightsSchedule[i].getDestination().equals(
    flightsSchedule[index].getDestination())&&
    flightsSchedule[index].getArrivalTime().before(
        flightsSchedule[i].getArrivalTime())&&
    flightsSchedule[index].addPassengers(num)){
System.out.println(flightsSchedule[index].getCodeF());
        return true;
    }
}
}

System.out.println("no such flight");
    return false;
}
    
```

**שאלה מס. 13**

s	if1	s.charAt(0)	s.charAt(1)
BANANA	T	A	B
ANANA	F	N	A
AANA	F	A	A
ANA	F	N	A
AA	F	A	A
A			

1. תוצאת הקריאה היא A. התו הקטן ביותר.

i	s.charAt(i)	c
0	B	N
1	A	
2	N	

2. תוצאת הקריאה היא BAANA. מחיקת המופע הראשון של התו המתקבל.

s	if1	s.charAt(0)	s.charAt(1)
BANANA	T	A	B
ANANA	F	N	A
AANA	F	A	A
ANA	F	N	A
AA	F	A	A
A			

3. תוצאת הקריאה AAABNN. הפעולה מחזירה מחרוזת המכילה את תוי המחרוזת המתקבלת בסדר עולה.

s	ans	a
BANANA		
BNANA	A	A
BNNA	AA	A
BNN	AAA	A
NN	AAAB	B
N	AAABN	N
	AAABNN	N

**שאלה מס. 14**

1. פעולת עזר המקבלת מטריצה ומקומים  $i, j$  ובודקת האם אין אף שכן (מתוך שמונת השכנים) שערכו זהה לערך התא במקום ה-  $i, j$ . אם כן, מוחזר אמת, אחרת מוחזר שקר.

```
public static boolean isSad (int [][]arr, int i, int j)
{
    int cnt=0;
    for (int x=i-1; x<i+1;x++)
        for (int y=j-1; y<j+1 ; y++)
            if (arr[x][y] ==arr[i][j])
                cnt++;
    return cnt==0;
}

public static boolean isSadInArray(int [][] mat, int num)
{
    for(int i=1; i<mat.length; i++)
        for (int j=1; j<mat[i].length-1; j++)
            if (mat[i][j]==num)
                if (isSad(mat, i, j))
                    return true;
    return false;
}
```

2. פעולת עזר המקבלת מערך חד ממדי של ערכים שלמים, גודל מערך ומספר שלם ומחזירה אמת אם הערך שהתקבל נמצא במערך, אחרת מוחזר שקר.

```
public static boolean notExistInArr(int [] vec, int index, int
num)
{
    int i=0;
    while (i<index &&vec[i]!=num) i++;
    return i==index;
}
```



```

        public static int [] allSadNumbers(int [][] mat){
            int index =0;
            int [] vec= new int [mat.length-2 * mat[0].length-2];
            for(int i=1; i<mat.length; i++)
                for (int j=1; j<mat[i].length-1; j++)
                    if (isSad(mat, mat[i][j]))
                        {
                            if (notExistInArr(vec, index, mat[i][j]))
                                {
                                    vec[index++]=mat[i][j];
                                }
                        }
                    if (index>0){
                        int [] arr = new int [index];
                        for(int i;i<index i++)
                            arr[i]=vec[i];
                        return arr;
                    }
                    else
                        return null;
                }
    
```

3.

```

        public static void printBigNum(int [][]mat)
        {
            int [] arr = allSadNumbers(mat);
            if (arr!=null){
                int max=0;
                for (int i=0;i<arr.length; i++)
                    if (arr[i]>=10 && arr[i]<=99)
                        max=Math.max(max, arr[i]);
                if (max!=0)
                    System.out.println(max);
            }
            else
    
```

```
System.out.println("no such number");  
  
    }  
    else  
        System.out.println("no sad numbers");  
    }
```