בית ספר אפי ארזי למדעי המחשב

המרכז הבינתחומי הרצליה

#### מבוא למדעי המחשב

#### מבחן אמצע + פתרון (דוגמא)

הנחיות:

* זמן המבחן: שעתיים. לא תהיה הארכה.
* זמן הבחינה מוגבל, ויש לעבוד ביעילות. אם נתקעים בסעיף מסוים, מומלץ לעזוב אותו ולרוץ הלאה.
* חומר סגור. השימוש במחשבים או במחשבונים אסור.
* ענו על כל השאלות על טופס המבחן. מומלץ להשתמש במחברות הבחינה כטיוטא. ניתן גם לכתוב תשובות במחברת הבחינה ולהפנות אליהן בצורה ברורה ומפורשת מטופס המבחן. חובה למחוק באופן ברור כל דבר שאינכם רוצים שנבדוק.
* בגמר הבחינה יש להחזיר את טופס המבחן + מחברת המבחן + כל דפי העזר הנלווים.
* אם תרגישו צורך לעשות הנחה מסוימת כדי לענות על שאלה מסוימת, ניתן לעשות זאת, כל עוד ההנחה היא סבירה ומנוסחת בדיוק ובבהירות.
* אם אתם לא מסוגלים לתת תשובה מלאה לשאלה מסוימת, תנו תשובה חלקית. תשובה נכונה באופן חלקי תקבל ניקוד חלקי.
* התשובות חייבות להיות קצרות ולעניין. כתב היד חייב להיות קריא וברור. תשובות לא קריאות תקבלנה ציון 0.
* אם התבקשתם לכתוב תכנית שאמורה לפעול על קלט מסוים, אזי התכנית לא אמורה לבדוק אם הקלט תקין, אלא אם כן נאמר כך בשאלה במפורש.
* התכניות שתכתבו תישפטנה, בין היתר, לפי האורך והאלגנטיות שלהן. תוכניות ארוכות או מסורבלות ללא צורך יקבלו פחות נקודות, אפילו אם הן ממלאות את המשימה שהוגדרה בשאלה.
* שימו לב: בשאלות הבאות אתם נדרשים לממש מתודות. אם אתם לא מצליחים מסיבה כלשהי לכתוב את הקוד הנדרש, ניתן לתאר במילים את המימוש שהייתם רוצים לבצע. אם התיאור יהיה מדויק ולעניין, תקבלו ניקוד חלקי.
* כל החומרים להם תזדקקו במהלך המבחן מתועדים בדפי העזר שקיבלת.
* לא יורדו נקודות על טעויות סינטקס טריוויאליות.

#### בהצלחה!

1. התבוננו במתודה הבאה. המתודה מקבלת מערך דו-ממדי שמייצג מטריצה ריבועית (מטריצה שמספר השורות שלה שווה למספר העמודות), ומחשבת משהו.

// This method receives a square matrix

// (in which the number of rows and columns is the same)

// and computes and returns something. What does this method do?

public static boolean mystery(int[][] m) {

int n = m.length-1;

int a1 = 0, a2 = 0;

for (int i = 0; i <= n; i++) {

a1 = a1 + m[i][i];

a2 = a2 + m[i][n-i];

}

return a1 == a2;

}

א. (10 נקודות) מה המתודה מחשבת? איזה ערך היא מחזירה? יש לתת תשובה כללית, כגון: "המתודה מחשבת את מספר העמודים במטריצה שאין בהם 0". כמובן שהמתודה הזאת עושה משהו כללי אחר. מה היא עושה? כתבו תשובה קצרה של לא יותר משני משפטים:

תשובה: המתודה מחשבת את סכום האיברים באלכסון הראשי וסכום האברים באלכסון המשני של המטריצה. אם הסכומים שווים, המתודה מחזירה true; אחרת, המתודה מחזירה false.

ב. (5 נקודות) המתודה הנ"ל מניחה, אך לא בודקת, שבמערך הנתון מספר השורות שווה למספר העמודים. מה יקרה אם נריץ את הקוד הנ"ל על מערך שבו מספר השורות אינו שווה למספר העמודים? (בהנחה שלא שינינו כלל את הקוד)? כתבו תשובה קצרה ומדויקת של לא יותר משני משפטים:

תשובה:

אם יש יותר שורות מעמודים, המתודה עושה מה שתואר בתשובה לסעיף א'.

אם יש פחות שורות מעמודים, הרצת המתודה תגרום ל- run-time error.

כל השאלות הבאות במבחן עוסקות במחלקת Sets, שמתועדת בדפי העזר.

מחלקת Sets היא ספרייה שמספקת שירותים שונים לביצוע פעולות על קבוצות שמכילות מספרים שלמים. קבוצה (set) מוגדרת כאוסף של ערכים (elements) ללא חזרות וללא סדר. לדוגמא, אוסף המספרים 3, 6, 3, 2, 1, 2 אינו קבוצה, כי לפחות אחד הערכים בו מופיע פעמיים. לעומת זאת,   
 { 3, 6, 2, 1 } היא קבוצה, כי אין בה חזרות. כמו כן, שתי הקבוצות { 3, 6, 2, 1 } ו- { 1, 2, 3, 6 } שוות זו לזו, כי סדר האברים בקבוצה הוא חסר משמעות.

אנו ממליצים להקדיש כעשר דקות לקריאת שני דפי העזר שכותרתםThe Sets Class עכשיו, לפני שעונים על השאלות הבאות.

ענו על שאלה 2 או על שאלה 3

2. (13 נקודות) המתודה print מקבלת מערך ומדפיסה אותו כקבוצה. לדוגמא, אם המערך הוא  
[7, 2, 1, 5], המתודה מדפיסה { 7, 2, 1, 5 }. כיתבו את המתודה.

/\*\* Prints the array as a set.

\* For example, if array is [7, 2, 1, 5], prints { 7, 2, 1, 5 } \*/

public static void print(int[] set) {

// Write your code here:

תשובה:

public static void print(int[] set){

System.out.print("{");

for (int i = 0; i < set.length; i++) {

System.out.print(" " + set[i]);

}

System.out.println(" }");

}

3. (13 נקודות) המתודה elementOf בודקת אם ערך נתון נמצא במערך נתון. לדוגמא, אם המערך הוא  
 [7, 2, 5, 1] והערך הוא 2, המתודה מחזירה true. אם הערך הוא 3, המתודה מחזירה false. כיתבו את המתודה.

/\*\* Checks if the given element exists in the given set. \*/

public static boolean elementOf(int e, int[] set) {

// Write your code here:

תשובה:

public static boolean elementOf(int e, int[] set) {

for (int i = 0; i < set.length; i++) {

if (e == set[i]){

return true;

}

}

return false;

}

ענו על שאלה 4 ועל שאלה 5

4. (10 נקודות) המתודה random מחזירה ערך שנבחר באופן מקרי ממערך. לדוגמא, אם המערך הוא  
 [3, 5, 1], המתודה יכולה להחזיר 5 בהרצה אחת, 3 בהרצה הבאה, 5 בהרצה הבאה, 1 בהרצה הבאה, 3 בהרצה הבאה, וכן הלאה. כיתבו את המתודה.

/\*\* Returns a random element chosen from the given set. \*/

public static int random (int[] set) {

// Write your code here:

תשובה:

public static int random (int[] set) {

// Computes a random index of the right size

int r = (int) (Math.random() \* (set.length - 1));

return set[r];

}

5. (25 נקודות) המתודה buildSet מקבלת מערך ומחזירה מערך שמכיל את כל המספרים במערך הנתון, ללא חזרות. לדוגמא, אם המערך הנתון הוא [ 3, 6, 3, 2, 1, 2 ] המתודה מחזירה את המערך [ 3, 6, 2, 1 ]. כיתבו את המתודה.

/\*\* Builds a set from a given array.

\* For example, if the given array is [7, 2, 7, 1, 5, 1],

\* the method returns the array [7, 2, 1, 5]. \*/

public static int[] buildSet (int[] arr) {

// Write your code here:

תשובה:

public static int[] buildSet (int[] arr) {

// Creates a temp array that contains only the unique elements

int[] temp = new int[arr.length];

int size = 0;

for (int i = 0; i < arr.length; i++) {

boolean found = false;

for (int j = 0; j < size; j++) {

found = found || arr[i] == temp[j];

}

if (!found) {

temp[size] = arr[i];

size++;

}

}

// Creates and returns an array of the right size.

int[] set = new int[size];

for (int i = 0; i < size; i++) {

set[i] = temp[i];

}

return set;

}

ענו על שאלה 6 או על שאלה 7

6. (25 נקודות) המתודה union מחזירה את האיחוד של שני מערכים שמייצגים קבוצות. האיחוד של שתי קבוצות הוא קבוצה שמכילה את כל האברים של הקבוצה הראשונה, ואת כל האברים של הקבוצה השנייה, ללא חזרות. לדוגמא, אם שני המערכים הם [ 2, 3, 5 ] ו- [ 5, 7, 2, 6 ] , המתודה מחזירה את המערך [ 5, 7, 2, 6, 3 ]. כיתבו את המתודה.

/\*\* Returns the union of two sets. \*/

public static int[] union (int[] set1, int[] set2) {

// Write your code here:

תשובה:

public static int[] union (int[] set1, int[]set2) {

// Creates a temporary array whose size is the maximal size of the union

int[] temp = new int[set1.length + set2.length];

// Adds the elements of the first set

for (int i = 0; i < set1.length; i++) {

temp[i] = set1[i];

}

int size = set1.length;

// Adds the elements of the second set that are not members of the first set

for (int i = 0; i < set2.length; i++) {

if (!elementOf(set2[i], set1)){

temp[size] = set2[i];

size++;

}

}

// Copies the temporary array to a new array that has the right size

int[] union = new int[size];

for (int i = 0; i < size; i++) {

union[i] = temp[i];

}

return union;

}

Another possible implementation (more elegant):

/\*\*

\* Returns the union of two sets.

\* This implementation uses the BuildSet function.

\*/

public static int[] union (int[] set1, int[]set2) {

int[] temp = new int[set1.length + set2.length];

for (int i = 0; i < temp.length; i++) {

if (i < set1.length){

temp[i] = set1[i];

} else {

temp[i] = set2[i - set1.length];

}

}

return buildSet(temp);

}

7. (25 נקודות) המתודה intersectionמחזירה את החיתוך של שני מערכים שמייצגים קבוצות. החיתוך של שתי קבוצות הוא קבוצה שמכילה את כל האברים שנמצאים גם בקבוצה הראשונה וגם בקבוצה השנייה. לדוגמא, אם שני המערכים הם [ 2, 3, 5 ] ו- [ 5, 7, 2, 6 ] , המתודה מחזירה את המערך [ 2, 5 ]. כתבו את המתודה.

/\*\* Returns the intersection of two sets. \*/

public static int[] intersection (int[] set1, int[] set2) {

// Write your code here:

תשובה:

public static int[] intersection (int[] set1, int[] set2) {

// Creates a temporary array whose size

// is the size of the smaller set

int[] temp = new int[Math.min(set1.length, set2.length)];

int size = 0;

int i;

for (i = 0; i < set1.length; i++) {

if (elementOf(set1[i], set2)) {

temp[size] = set1[i];

size++;

}

}

// Creates and returns an array of the right size

int[] intersection = new int[size];

for (int j = 0; j < intersection.length; j++) {

intersection[j] = temp[j];

}

return intersection;

}

(סוף המבחן)The Sets class (help sheet given as-is together with the exam)

/\*\*

\* This class is a library of various operations on sets that contain integer values.

\* A set is a collection of values without repetition.

\* For example, 2,5,2,7 is not a set. {2,5,7} is a set.

\* The order of the elements in the set is insignificant.

\* For example, {2, 5, 7} is the same set as {5, 7, 2}.

\*

\* In this implementation sets are represented as arrays of integers.

\* In particular, a set that contains n elements is implemented by an

\* int[] array of size n.

\*/

public class Sets {

public static void main(String[] args){

// Creates two sets using the buildSet function

int[] a1 = { 2, 5, 3, 2, 7, 5 };

int[] s1 = *buildSet*(a1);

System.*out*.print("s1 = ");

*print*(s1);

int[] a2 = { 7, 5, 5, 6, 7 };

int[] s2 = *buildSet*(a2);

System.*out*.print("s2 = ");

*print*(s2);

// Tests the elementOf function

System.*out*.println("5 is an element of s1: " + *elementOf*(5, s1));

System.*out*.println("8 is an element of s1: " + *elementOf*(8, s1));

// Prints the intersection of s1 and s2

System.*out*.print("Intersection of s1 and s2 = ");

*print*(*intersection*(s1,s2));

// Prints the union of s1 and s2

System.*out*.print("Union of s1 and s2 = ");

*print*(*union*(s1, s2));

// Tests the random function

System.*out*.println("A random value drawn from the set s1: " + *random*(s1));

}

// Other method definitions are given in the next page.

}

The output:

% javac Sets.java

% java Sets

s1 = { 2 5 3 7 }

s2 = { 7 5 6 }

5 is an element of s1: true

8 is an element of s1: false

Intersection of s1 and s2 = { 5 7 }

Union of s1 and s2 = { 2 5 3 7 6 }

A random value drawn from the set s1: 5

The Sets class (help sheet given as-is together with the exam)

/\*\*

\* This class is ... (same documentation as in previous page)

\*/

public class Sets {

public static void main(String[] args){

// Same main method as in previous page

}

/\*\* Prints the array as a set.

\* For example, if the array is [7, 2, 1, 5], prints { 7, 2, 1, 5 } \*/

public static void print(int[] set) {

}

/\*\* Checks if the given element exists in the given set. \*/

public static boolean elementOf(int e, int[] set) {

}

/\*\* Returns a random element chosen from the given set. \*/

public static int random (int[] set) {

}

/\*\* Builds a set from a given array.

\* For example, if the given array is [7, 2, 7, 1, 5, 1],

\* returns the array [7, 2, 1, 5]. \*/

public static int[] buildSet (int[] arr) {

}

/\*\* Returns the union of two sets. \*/

public static int[] union (int[] set1, int[] set2) {

}

/\*\* Returns the intersection of two sets. \*/

public static int[] intersection (int[] set1, int[] set2) {

}

} // end of the Sets class.

Implementation tips:

* The methods of class Sets do not modify their arguments.
* To keep things simple, you don’t have to check if the received arrays represent sets. In other words, you don’t have to check if they contain repetition. The exception is the method buildSet, whose entire purpose is to remove repetitions.

Math Library API (help sheet given as-is together with the exam)

If you need to use a mathematical function, you may find this page helpful (it contains many more functions than you actually need for this exam).

|  |  |
| --- | --- |
| static int | [**abs**](http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#abs(int))(int a)  Returns the absolute value of an int value. |
| static double | [**log**](http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#log(double))(double a)  Returns the natural logarithm (base *e*) of a double value. |
| static double | [**log10**](http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#log10(double))(double a)  Returns the base 10 logarithm of a double value. |
| static int | [**max**](http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#max(int,%20int))(int a, int b)  Returns the greater of two int values. |
| static int | [**min**](http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#min(int,%20int))(int a, int b)  Returns the smaller of two int values. |
| static double | [**pow**](http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#pow(double,%20double))(double a, double b)  Returns the value of the first argument raised to the power  of the second argument. |
| static double | [**random**](http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#random())()  Returns a double value with a positive sign, greater than or  equal to 0.0 and less than 1.0. |
| static long | [**round**](http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#round(double))(double a)  Returns the closest long to the argument, with ties rounding up. |
| static double | [**sqrt**](http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#sqrt(double))(double a)  Returns the correctly rounded positive square root of a double value. |