Efi Arazi School of Computer Science

Introduction to Computer Science

Midterm Examination Sample + Solution

* The exam lasts 2 hours. There will be no time extension.
* Use your time efficiently. If you get stuck somewhere, leave the question and move on to another question.
* Use of digital devices, books, lecture notes, and anything other than this exam form is forbidden. All the materials that you need for answering this exam are supplied with the exam.
* Answer all questions on the current exam form.
* Answer all the questions on the exam pages. **Don’t write anything on the back of the pages**. Only the front pages are scanned for grading. You will get some blank pages for draft (טיוטה).
* You can answer any question in either English or Hebrew.
* If you feel a need to make an assumption, you may do so as long as the assumption is reasonable and clearly stated.
* If you can't give a complete answer, give a partial answer. A partial answer will award partial points.
* If you are asked to write code and you feel that you can't write it, you may describe what you wish to do in natural language (English or Hebrew). A good explanation will award partial credit.
* If you are asked to write code that operates on some input, there is no need to validate the input unless you are explicitly asked to do so. Likewise, if you are asked to write a function that operates on some arguments, there is no need to validate the arguments unless you are explicitly asked to do so.
* There is no need to document the code that you write, unless you want to communicate something to us.
* The code that you will write in the exam will be judged, among other things, on its conciseness, elegance, and efficiency. Unnecessarily long or cumbersome code will cause loss of points, even if it provides the correct answer.
* No points will be taken for trivial syntax errors. For example, instead of writing System.out.println(x) you can write println(x).

Good Luck!

A note about exam and homework solutions: Every programming question can be answered in more than one way. The solutions shown here are just one example. Other correct solutions are possible, and can potentially receive full points also.

Consider the following program:

// This function receives a square matrix (in which the number of rows and columns is the same)

// and computes and returns something.

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;

}

1. (8 points) What does this function compute? What value does it return? We expect you to give a general answer, such as: “The function computes and returns the number of columns in the matrix which do not contain a zero”. Of course, this function does something else. What does it do? Write a short answer of no more than two sentences.

Answer: The function computes the sum of the elements along the main diagonal (אלכסון ראשי) and the sum of the elements along the secondary diagonal (אלכסון משני). If these two sums are equal, the function returns true; otherwise, the function returns false.

2. (5 points) The function above assumes, but does not check, that the given array has an equal number of rows and columns. What would happen if we run this code with an array for which the number of rows is different from the number of columns? Write a short and precise answer of no more than two sentences.

Answer: if the array has more rows than columns, there will be a run-time error. If the array has fewer rows than columns, the function will work, but will operate on a submatrix of the given matrix.

All the remaining questions deal with the class Sets, described in a help page.

The Sets class is a library providing various services for performing operations on sets (**קבוצות**) containing integer numbers. A set is defined as a collection of elements without regard to order and without duplicate values. For example, the collection 3, 6, 3, 2, 1, 2 is not a set, since at least one of the values appears more than once. In contrast, {3, 6, 2, 1} is a set. The sets {3, 6, 2, 1} and {1, 2, 3, 6} are considered equal, since the order of the elements in a set is not significant

**Before answering or even reading the remaining questions, spend about ten minutes reading “The Sets Class” help page.**

3. (10 points) Implement the following function.

/\*\* Prints the elements of the set, starting and ending with curly brackets.

\* For example, if the set contains the elements 7, 2, 1, prints { 7, 2, 1 } \*/

public static void print(int[] set) {

// Write your code here:

Answer:

public static void print(int[] set) {

int n = set.length;

System.out.print("{");

for (int i = 0; i < (n - 1); i++) {

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

}

System.out.println(" " + set[n - 1] + " }");

}

4. (8 points) Implement the following function.

/\*\* Checks if the value e appears in the set.

\* For example, if the set is { 7, 2, 5 } and the value is 2, returns true.

\* if the set is { 7, 2, 5 } and the value is 3, returns false. \*/

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

// Write your code here:

Answer:

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

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

if (e == set[i]) {

return true;

}

}

return false;

}

**Note:** Assume that all the Sets functions described in this exam have been implemented correctly.

The functions that you implement can call any of these functions, if you think that it makes sense.

5. (8 points) Implement the following function.

/\*\* Checks if set1 is a subset (תת-קבוצה) of set2.

\* Set1 is a subset of set2 if every element of set1 is an element of set2. \*/

public static boolean subsetOf(int[] set1, int[] set2) {

// Write your code here:

Answer:

public static boolean subsetOf(int[] set1, int[] set2) {

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

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

return false;

}

}

return true;

}

6. (15 points) Implement the following function.

/\*\* Appends (adds) the value e to the set.

\* If e exists in the set, returns the set (a set cannot have duplicate values).

\* Otherwise returns a set containing all the elements of the set, and e. \*/

public static int[] append(int[] set, int e) {

// Write your code here:

Answer:

public static int[] append(int[] set, int e) {

if (elementOf(e, set)) {

return set;

}

int[] newSet = new int[set.length + 1];

for (int i = 0; i < (newSet.length - 1); i++) {

newSet[i] = set[i];

}

newSet[newSet.length - 1] = e;

return newSet;

}

7. (23 points) Implement the following function. Don’t use the append function, since it is inefficient.

/\*\* Returns the union (איחוד) of the two sets.

\* The union of set1 and set2 is the set containing all the elements of set1

\* and all the elements of set2 that are not elements of set1. \*/

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

// Write your code here:

Answer:

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

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

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

// Adds the elements of set1

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

temp[i] = set1[i];

}

int size = set1.length;

// Adds the elements of set2 that are not members of set1

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

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

temp[size] = set2[i];

size++;

}

}

// Creates and returns 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;

}

8. (23 points) Implement the following function. Don’t use the append function, since it is inefficient.

/\*\* Returns the intersection (חיתוך) of the two sets.

\* The intersection of set1 and set2 is the set containing all the elements of set1

\* that are also elements of set2. \*/

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

// Write your code here:

Answer:

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

\* The intersection of set1 and set2 is the set containing all the elements of set1

\* that are also elements of set2. \*/

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)];

// Adds the elements of s1 that are also elements of s2

int size = 0;

for (int 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;

}

(End of Exam, Help pages follow)

The Sets class (this page is supplied together with the exam)

/\*\* This class features operations on sets that contain integer values.

\* In this class, 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 and prints three sets

int[] s1 = { 2, 5, 3, 7 };

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

int[] s2 = { 7, 9, 2 };

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

int[] s3 = { 5, 7 };

System.out.print("s3 = "); println(s3);

// 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));

// Tests the subsetOf function

System.out.println("s3 is a subset of s1: " + subsetOf(s3, s1));

System.out.println("s3 is a subset of s2: " + subsetOf(s3, s2));

// Tests the append function

System.out.print("The set resulting from s2 and 9 appended: "); println(append(s2, 9));

System.out.print("The set resulting from s2 and 5 appended: "); println(append(s2, 5));

// Tests the union function

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

println(union(s1, s2));

// Tests the intersection function

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

println(intersection(s1, s2));

}

// Following the main function come all the functions listed in the exam questions.

}

Here is what happens when we compile and execute this program:

% **javac Sets.java**

% **java Sets**

s1 = { 2, 5, 3, 7 }

s2 = { 7, 9, 2 }

s3 = { 5, 7 }

5 is an element of s1: true

8 is an element of s1: false

s3 is a subset of s1: true

s3 is a subset of s2: false

The set resulting from s2 and 9 appended: { 7, 9, 2 }

The set resulting from s2 and 5 appended: { 7, 9, 2, 5 }

Union of s1 and s2 = { 2, 5, 3, 7, 9 }

Intersection of s1 and s2 = { 2, 7 }

|  |  |
| --- | --- |
| Diagram, venn diagram  Description automatically generated  Union | Venn diagram  Description automatically generated  Intersection |

Math Library API (this page is supplied 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**(int a)  Returns the absolute value of an int value. |
| static double | **log**(double a)  Returns the natural logarithm (base *e*) of a double value. |
| static double | **log10**(double a)  Returns the base 10 logarithm of a double value. |
| static int | **max**(int a, int b)  Returns the greater of two int values. |
| static int | **min**(int a, int b)  Returns the smaller of two int values. |
| static double | **pow**(double a, double b)  Returns the value of the first argument raised to the power of the second argument. |
| static double | **random**()  Returns a double value, greater than or equal to 0.0 and less than 1.0. |
| static long | **round**(double a)  Returns the closest long to the argument, with ties rounding up. |
| static double | **sqrt**(double a)  Returns the correctly rounded positive square root of a double value. |

A for loop example

In case you forgot the syntax, here is an example of a for loop that sums up the values of an array:

int sum = 0;

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

sum = sum + arr[i];

}