דגשים למבחן מה"ט-אלגוריתמיקה ותכנות

באופן כללי:

- שימי לב לשעון, חלקי את הזמן בצורה נכונה בין השאלות
- כתבי תיעוד לכל קוד שאת כותבת
- שימי לב להוראות ותקראי אותן בדיוק רב
- אם את כותבת את פקודות הקלט/פלט בקיצור-בראש המבחן תכתבי מקראה
- בעמוד הראשון במחברת הבחינה לא לכתוב פתרונות-אלא את מס' השאלות עליהן בחרת לענות

מחרוזות

ניתן להתייחס למחרוזת כאל מערך תווים.
גישה לתו במחרוזת, כמו לתא במערך. לדו': str[2] כאשר str="hello" ייצג את האות l.
הפעולות המוגדרות במחלקת string הן פעולות פנימיות (IndexOf/ToUpper/SubString וכו') ולכן הגישה אליהן תהיה: str.SubString(2) לדו' המחרוזת. שם הפעולה.שם(...)
שימי לב: פעולות אלו אינן משנות את המחרוזת המקורית עליה הופעלה הפעולה!
מיקום התו הראשון במחרוזת: 0
מיקום התו האחרון במחרוזת: str.Length-1
(מצורף קובץ סיכום פעולות על מחרוזות)

מערכים

לא ניתן להכניס איברים למערך טרם הוגדר לו גודל. (כמובן כשמקבלים מערך בפעולה או כשכבר הוגדר במחלקה וכו' ניתן להתייחס למערך כביכול יש לו גודל מוגדר מראש-למעט אם קיימת הוראה שונה)
כל תאי המערך מאותחלים בעת הגדרתו (טרם הוכנסו ערכים) ע"פ הטיפוס שלו.
int-מאותחל באפסים
double-מאותחל באפסים
char-מאותחל ל \0 (שום דבר, כשנדפיס על המסך לא נראה דבר)
string-מאותחל ל null (גם כאן, כשנציג על המסך לא נראה דבר)
bool-מאותחל ל false
עצמים-מאותחל ל null
מיקום התו הראשון במערך: 0
מיקום התו האחרון במערך: arr.Length-1 <-שם המערך
יש לשים לב לא לחרוג מגבולות המערך! לדו':
בעת גישה בלולאה למיקום ה i-1, לשים לב להתחיל את סריקת הלולאה מ 1.
בעת גישה בלולאה למיקום ה i+1, לשים לב לסיים את סריקת הלולאה ב arr.Length-1 וכו'

מטריצות-מערך דו מימדי

לא ניתן להכניס איברים למערך טרם הוגדר לו גודל. (כמובן כשמקבלים מערך בפעולה או כשכבר הוגדר במחלקה וכו' ניתן להתייחס למערך כביכול יש לו גודל מוגדר מראש-למעט אם קיימת הוראה שונה)
מס' איברי המטריצה: mat.Length
מס' השורות במטריצה: mat.GetLength(0)
מס' העמודות במטריצה: mat.GetLength(1)
מס' העמודות=כמה איברים יש בכל שורה

מס' השורות=כמה איברים יש בכל עמודה
לכן:
בסריקת שורה ספציפית, נעבור בלולאה ע"פ מס' העמודות, כשהשורה קבועה. לדו' mat[2,i]
בסריקת עמודה ספציפית, נעבור בלולאה ע"פ מס' השורות, כשהעמודה קבועה. לדו' mat[i,3]

סיבוכיות

בעת חישוב סיבוכיות, נחפש בקוד את הקריאה לפעולות, ואת הלולאות.
נבדוק לכל אחד מהם מה סיבוכיות זמן הריצה שלו. (תמיד נתחיל עם החישוב הפנימי ביותר)
שימי לב!

- קשר של + מתבטל, קשר של * נשאר.
- לולאה שעוברת מס' קבוע של פעמים=O(1)

תכנות מונחה עצמים

גישה לתכונה בתוך המחלקה-ישירות באמצעות שמה
גישה לתכונות מחוץ למחלקה בה הוגדרו (גם כשמדובר במחלקה מכילה!)-באמצעות פעולות גישה Get/Set (במידה ולא מוגדרות פעולות גישה שיש בהן צורך-הוסיפי אותן בעצמך)
פעולות בונות-לשים לב לנתוני השאלה, גם כשלא כתוב לאתחל ערכים לתכונות מסוימות-אם ניתן להבין למה לאתחל אותן (לדו' גודל מערך כשנתון הגודל) יש לאתחל.
זימון פעולה פנימית-שם הפעולה.שם העצם. ניתן לזמן פעולה פנימית רק על עצמים מטיפוס המחלקה בה הוגדרה הפעולה!
שינוי ערך לתכונה מחוץ למחלקה-לעולם לא a.GetName()="ploni" אלא a.SetName("ploni");
מערך עצמים
לפני גישה לתכונות עצם בצערך-יש לוודא שבאותו תא קיים עצם ואינו ריק (מצביע ל null)
לכן נבדוק if(arr[i]!=null) ורק אח"כ ניגש לתכונות העצם (אלא אם נתון כי המקומות מלאים)

פעולה המשנה משתנים מטיפוסים מורכבים-מערכים/מטריצות/עצמים

בשונה מ 5 הטיפוסים הפשוטים-int/double/bool/char/string משתנים מטיפוסים מורכבים כמו מערכים/מטריצות/עצמים-כשמשנים אותם בפעולה הם משתנים גם מחוץ לפעולה.
לדו', אם נשלח את המערך הבא {5,7,12,3} לפעולה המגדילה ערך כל איבר במערך ב 1, לאחר הפעולה המערך ישתנה ל {6,8,13,4}.
אך אם נשלח את המשתנה num=5 לפעולה המוסיפה אחד לפרמטר שהיא מקבלת, לאחר הפעולה כשנציג את num נראה כי ערכו עדיין 5…
(השינוי על משתנים מטיפוסים פשוטים חל רק בתוך הפעולה)

טענת כניסה וטענת יציאה

טענת כניסה-ההנחה לגבי המצב לפני הפעלת הפעולה. לדו': הקלט הוא מערך של מספרים שלמים, בגודל לפחות 1.
טענת יציאה-ההבטחה לגבי מה שהפעולה תעשה אחרי שהיא רצה. לדו': הפעולה מחזירה את המספר הגדול ביותר במערך.

Random מספרים אקראיים

יש להגדיר פעם אחת משתנה מטיפוס Random כדי שנוכל לבצע באמצעותו "הגרלות". כך:
Random rnd=new Random();

ואח"כ נשתמש ב rnd כדי להגריל מספרים, באמצעות הפעולה Next. ניתן להשתמש בה ב 2 צורות:

1. הגרלת מס' בטווח מסוים: rnd.Next(10,101) יגריל מס' בין 10 ל 100 (לא כולל גבול עליון)
2. הגרלת מס' מ 0 עד מס' מסוים: rnd.Next(101) יגריל מס' בין 0 ל 100 (כנל')

פעולות מתמטיות מוגדרות בשפה-מחלקת Math

פעולות חיצוניות, זימון הפעולות: (...)שם הפעולה.Math (מצורף קובץ פעולות מחלקת Math)

המרה

המרה מ Int ל double ולהפך, בעזרת (int) או (double)

לדו':

```
int num=5;
double num2=(double)num;
```

(שימושי בעיקר בהמרת פרמטר בתרגיל חילוק בין שלמים לעשרוני כדי לקבל תוצאה מדויקת)

המרה מ char ל int ולהפך, בעזרת (char) או (int)

שימי לב!

בהמרת תו למספר שלם, התו יומר לערכו בקוד ASCII.

המרה מכל טיפוס למחרוזת, באמצעות ()ToString.שם המשתנה

המרה מ string לכל טיפוס אחר:

(מחרוזת)Parse.הטיפוס אליו נרצה להמיר

רקורסיה

בכל קריאה לפעולה נפתח "קריאה חדשה"-תא חדש עבור הקריאה.

כשיפסקו הקריאות (בפעולה מסוג void)/כשיוחזר ערך-נחזור אחורה בקריאות כלפי מעלה.

בכל קריאה נוודא כי ביצענו את כל שורות הקוד שהיו אמורות להתבצע עד סיום הפעולה/ערך מוחזר. גם כאן, כשנמצא קריאה לפעולה נפתח "קריאה חדשה"

יש לציין בכל קריאה את הערכים הזמניים של כל משתנה.

שימי לב לכתוב את הערך המוחזר/מה מבצעת הפעולה בצורה מודגשת מחוץ למעקב הכללי.

טבלת מעקב

בכל מקרה בו יש לבצע מעקב (לא רקורסיבי- אם כי גם ברקורסיה ניתן לעקוב באמצעות טבלה) ולא מצוינות העמודות, נצייר טבלה עם העמודות הבאות:

עמודה לכל משתנה (כולל מערך, או arr[i] ->תא במערך)

עמודה לכל תנאי (כולל תנאי לולאות)

עמודה לפלט/ערך מוחזר במידה ויש

בטבלת מעקב לרקורסיה-עמודה ל"משפט זימון"

בכל שורת קוד נעבור לשורה הבאה בטבלה, למעט לולאות-בהן ניצור שורה לכל איטרציה בלולאה.

בהצלחה רבה!