



הוראות עבודה והגשה להגנה סמסטר ב' מועד א' תשע"ח פיתוח תוכנה מתקדם 1

כללי - המבחן בודק אוטומטית את הפרויקט וכולל 3 בדיקות עיקריות:

- בדיקת design (30 נק')
- בדיקת ריצה לשרת (40 נק')
- בדיקת ריצה לאלגוריתם ה Best First Search על בעיית חיפוש שונה (30 נק').

בבחינה זו עליכם לנתב את הבדיקות לפרויקט שלכם בקובץ TestSetter.java, להגיש אותו + PTM_projectTest.jar המכיל את הפרויקט שלכם למערכת הבדיקות ולתקן טעויות במידת הצורך. לשם כך יש לכם **4 שעות**.

הוראות:

1. היכנסו ל <https://ck.cs.colman.ac.il/test.jsp>, בחרו **PTM1_semester_B** ו **moed_a**, הקלידו את ת.ז שלכם (9 ספרות) ובחרו "הורד מטלה". תקבלו קובץ zip המכיל את התשתית לבדיקה.
2. **צרו מקומית פרויקט חדש** ובו package בשם test והעתיקו לשם את קובצי המקור מה zip. מעתה נפנה לפרויקט כ "פרויקט הבדיקה"
3. צרפו את ה PTM_projectTest.jar לפרויקט הבדיקה, או
 - a. מומלץ יותר, לפתוח את הפרויקט שלכם באותה סביבה ולצרף אותו ישירות, כך שאם תידרשו לתיקונים בפרויקט שלכם תוכלו לעשות זאת ישיר ולא בכל פעם לעדכן את ה jar.
 - b. באקליפס זה דרך add.. -> projects -> configure -> build path
 - c. **באף שלב אין לשנות או להוסיף קוד לפרויקט זה**. הקוד היחיד שיש לכתוב הוא בתוך TestSetter.java
4. בפרויקט הבדיקה התבוננו ב TestSetter.java. כאן אתם מנתבים את הבדיקה לפרויקט שלכם. הדבר היחיד שהשתנה ממוד האימון שנחשף לכם עוד בסמסטר זה
 - a. העדכון של טיפוס ה Searchable, Searcher והמימושים שלהם במתודה setClasses
 - b. המתודה solveWordGame שבה תפתרו באמצעות Best First Search את המשחק Word Game. יוסבר בעמוד הבא כיצד.הקוד הכתוב ב TestSetter **הוא רק לדוגמא**. במתודה setClasses התאימו את המחלקות השונות. במתודה runServer הפעילו את השרת שלכם, **וודאו שהוא רץ בת'רד נפרד** כדי לא לתקוע את המערכת. במתודה stopServer עצרו אותו, וכאמור במתודה solveWordGame עליכם לפתור משחק מילים כפי שיוגדר ויוסבר בעמוד הבא. **שימו לב! אין לשנות דבר בחתימות של TestSetter** **לרבות throws Exception ואין לשנות דבר בשאר הקבצים שקבלתם.**
5. התבוננו ב MainTrain.java. ניתן לראות את 3 הבדיקות העיקריות. השתיים הראשונות זהות למוד האימון שהכרתם עד כה, ואילו השלישית בודקת פתרון של Word Game.
6. תריצו את פרויקט הבדיקה לוקאלית עד שנמנעתם לחלוטין משגיאות קומפילציה וריצה, ונמנעתם ככל הניתן מהודעות שגיאה לוגיות שפולט ה MainTrain.
7. העלו למערכת הבדיקה את TestSetter.java ואת PTM_projectTest.jar. אם עדכנתם את הפרויקט שלכם, אל תשכחו לעדכן את ה jar! כמו כן, **וודאו שקוד המקור מוכל** בתוך ה jar.
8. כעת הגישו למוד הגשה ותתקנו בהתאם להודעות השגיאה של הבדיקות החסויות. שימו לב שהפעם נבדקים 4 שלבים **אקראיים** בגודל של 3X3 כך שעל כל חלק צינור שאינו במקומו יורדות 4 נקודות, ובסך מצטבר של 40 נקודות. אם השרת שלכם לא מחזיר פתרון מושלם אז בהחלט ייתכן הבדל בין הגשות שונות בגלל אקראיות השלבים. לכל שלב מובטח פתרון אחד ויחיד.
9. מרוצים מהציון? מעולה. תגישו **במוד הגשה סופית**. לאחר הגשה זו תקבלו **מס' אסמכתא** בן 4 ספרות שאותו עליכם להציג לבדיקת הבוחנות בטרם יציאתכם מהכיתה.
10. הגישו את כל הקבצים לשרת הגיבוי. התעלמו ממס' האסמכתא שניתן שם.
11. הציון ייקבע ע"פ הציון הגבוה ביותר שנצפה במוד הגשה.

מיותר לציין שאין להעתיק במבחן או לשלוח בת.ז שונה משלך או לצאת עם אסמכתא מומצאת וכו'. כל נסיון העתקה ייתפס ויש לכך השלכות מאד חמורות. נא לא להתעסק עם זה.

נספח: פתרון Word Game

במשחק פשוט זה ניתנת לנו מחרוזת עם אותיות מעורבבות. בכל צעד נוכל להחליף בין זוג אותיות עד שנגיע למחרוזת היעד.

לדוגמא: עבור מצב התחלתי של "bdca" ומצב יעד של "abcd" נרצה להחליף תחילה את אינדקסים 0,3 (נקבל בינתיים "adcb") ואז את אינדקסים 1,3 ונקבל את מחרוזת היעד.

המחיר להחליף כל שתי אותיות זה המרחק בין האינדקסים שלהם.

נתונה לכם המחלקה WordGame – **אין לשנות בה דבר**.

- למחלקה זו מגדירים בבנאי את המחרוזת ההתחלתית ואת מחרוזת היעד.
- היא מאפשרת החזרה של המחרוזת הנוכחית, האם היא זהה למטרה, ומהו המצב ההתחלתי.
- המתודה switchLetters מחליפה בין האינדקסים i,j ומחזירה את המרחק ביניהם.
 - היא תחזיר 1- אם הזנתם אינדקסים לא חוקיים.
- המתודה applyActions:
 - מצפה לקבל רשימה של מחרוזות, כל מחרוזת מהצורה "x,y" כאשר x,y מהווים ערכים של אינדקסים להחלפה. למשל "0,3".
 - עבור כל מחרוזת ברשימה המתודה מחליפה את האינדקסים שניתנו

הביטו בבדיקה ב MainTrain

- היא מגדירה מחרוזת התחלתית של "bdca" ומחרוזת מטרה של "abcd".
 - מפעילה את המתודה TestSetter.solveWordGame() שעליכם לממש, שמחזירה רשימה של מחרוזות מהצורה ש applyActions מצפה לקבל.
 - כעת היא מפעילה את applyActions ולאחר מכן בודקת שאכן הגענו למטרה.
- אופי הבדיקה ב MainTest זהה. הערכים כמובן שונים.

עליכם לממש את המתודה TestSetter.solveWordGame().

בפרט, עליכם להחליף שם את הקוד הספציפי בקוד שיוצר Searchable מתאים מתוך ה WorGame שקבלתם כפרמטר, להזין אותו לאלגוריתם ה BestFS, ולהחזיר את הפתרון בדמות רשימה של מחרוזות מהסוג "x,y".

דגש חשוב: כל האדפטציה צריכה לקרות בתוך המתודה הזו (השתמשו במופע אנונימי של Searchable). אין לשנות את קוד הפרויקט שהבאתם מהבית.

בהצלחה!