

## הצלחה

### הוראות להגשה

1. יש להגיש קובץ zip יחיד בעל השם EXP\_ID1\_ID2 כאשר ID1 ו ID2 הם מספרי תעודות הזהות של שני בני הזוג. קובץ ה zip יכול תיקייה בודדת בשם src ובה כל קבצי ה Java שיצרתם, ללא תיקיות נוספות ותתי תיקיות. **אין להגיש את קובץ ה-main.**
2. ההגשה תתבצע רק ע"י אחד מבני הזוג למקום הייעודי באתר הקורס במודל.
3. זוג שהתרגיל שלו לא יתקמפל או יעוף בזמן ריצה **ציונו בתרגיל יהיה 0.**
4. יש לכתוב קוד קריא ומסודר עם שמות משמעותיים למשתנים, למתודות ולמחלקות. יש לתעד את הקוד.
5. יש להקפיד למלא את כל דרישות התרגיל. **אי עמידה בדרישות התרגיל תגרור הורדת ציון ואף פסילת התרגיל.**

## הקדמה

בתרגיל זה תממשו בעצמכם בשפת java "רשת ראוטרים" ואת אלגוריתם link state routing אשר נלמד בהרצאה. בתחילת התוכנית כל ראوتر יחזיק את רשימת השכנים שלו, הפורטים באמצעותם הם מתקשרים, והמשקולות על הקשתות בין השכנים. במהלך ריצת התוכנית כל קודקוד בגרף (קרי ראوتر) יבנה מטריצת שכנויות של כל הגרף, באמצעות אלגוריתם link state routing, ויוכל לעדכן אותה בסבבים השונים. בנוסף כל קודקוד מסוגל לעדכן את המשקולות על הקשת שלו כאשר היא משתנה.

## המחלקות בתרגיל

1. **מחלקת Pair:** לנוחיותכם סופקה לכם מחלקה גנרית בשם Pair אשר מייצגת זוג. לכל זוג יש מפתח וערך, אם כי המשמעות שלהם נתונה לבחירתכם. המחלקה מממשת את הממשק Serializable, כלומר ניתן לשלוח אובייקטים של המחלקה ברשת. אין לשנות את המימוש של המחלקה הזו.
2. **מחלקת Node:** עליכם לממש מחלקה בשם Node אשר מייצגת קודקוד/ראوتر ברשת. על המחלקה לרשת מThread או לממש את Runnable (שכן כל הקודקודים מתקיימים "רשת" שאנו בונים בו זמנית). כאשר נקרא למתודה start של תהליך שמייצג קודקוד, הקודקוד יריץ את אלגוריתם link state routing. לכל קודקוד יש מזהה ייחודי בין 1 למספר הקודקודים בגרף (וכל קודקוד יודע גם מי השכנים שלו ומספר הקודקודים בגרף). בתחילת הריצה כל קודקוד קורא מקובץ הקלט את מזהי השכנים שלו והפורטים באמצעותם הוא מתקשר עם כל שכן (עבור כל שכן קיימים שני פורטים - פורט אליו הקודקוד שולח את ההודעות, ופורט ממנו הקודקוד מקבל את ההודעות). במהלך ריצת התוכנית על כל קודקוד ליצור את מטריצת השכנויות באופן הבא: השורה/עמודה ה-i מייצגת את הקודקוד בעל המזהה i+1. בתא j, יהיה 1- אם אין קשת בין שני הקודקודים, אחרת בתא תהיה המשקולת של הקשת. בנוסף על הקודקוד לממש פונקציה שאינה מחזירה ערך ולא מקבלת פרמטרים בשם print\_graph. הפונקציה מדפיסה את מטריצת השכנויות. את פורמט ההדפסה ניתן למצוא בקבצי הקלט.
3. **מחלקת ExManager:** בקבצי התרגיל סופק לכם קובץ ExManager אשר מכיל מחלקה באותו השם. תפקיד מחלקה זו היא ליצור ולנהל עבורנו את הרשת. למחלקה בנאי יחיד אשר מקבל את הנתיב לקובץ הקלט. בנוסף על המחלקה לממש את הפונקציות הבאות:
  - public Node getNode: הפונקציה מקבלת משתנה יחיד שהוא מזהה של קודקוד ועליה להחזיר את הקודקוד עם המזהה הזה.
  - public int getNum\_of\_nodes: אינה מקבלת פרמטרים ומחזירה את מספר הקודקודים בגרף.
  - public void update\_edge: הפונקציה מקבלת מזהים של שני קודקודים שיש ביניהם קשת ומשקולת, ואחראית לעדכן עבור כל אחד **משני הקודקודים** את ערך המשקולת החדש.
  - public void read\_txt: הפונקציה קוראת את קובץ הקלט עד השורה stop, ומייצרת את קודקודי הגרף עם המידע הרלוונטי לתחילת התרגיל (לפרטים נוספים ראו את מבנה קובץ הקלט).
  - public void start: הפונקציה מורה לגרף לבצע את אלגוריתם link state routing.אתם מוזמנים להוסיף פונקציות פנימיות לנוחיותכם אולם עליכם לממש את כל הפונקציות

הנ"ל. במהלך ריצת התרגיל לאחר כל איטרציה של האלגוריתם link state routing, קודקוד אקראי מתבקש להדפיס את מטריצת השכנויות שלו. בסיום כל ריצת link state routing כל מטריצות השכנויות אמורות להיות זהות.

**שימו לב! התקשורת בין שני קודקודים שכנים תתבצע אך ורק על ידי תקשורת ישירה בין הקודקודים בממשקי שרת לקוח אותם אתם נדרשים לממש. העברת הודעות בין קודקודים באמצעות מחלקת ExManager, או ערכי משקולות בקודקוד מסויים של קשתות שאינן נוגעות בו תגרור פסילה מיידית של התרגיל!**

4. **קובץ main:** קובץ הmain אחראי להריץ את התרגיל. הוא מייצר אובייקט ממחלקת ExManager ובאמצעותו מורה לרשת לבצע את אלגוריתם link state routing, לעדכן קשתות ולהדפיס את מטריצת השכנויות של קודקוד מסויים. בחינת התרגיל תעשה באמצעות קובץ דומה, לכן מומלץ לא לשנות את הקובץ.

הערה: אתם מוזמנים לממש ולהוסיף מחלקות משלכם.

### יבוא ספריות

בתרגיל זה מותר לבצע אך ורק את imports הבאים:

```
import java.net.*;
import java.util.*;
import java.io.*;
```

### מבנה קובץ הקלט

על המחלקה ExManager לקרוא את קובץ הקלט עד אשר היא מגיעה לשורה stop. כל השורות בקובץ עד שורה זו מכילות פרטים לבנית הרשת. הקובץ בנוי באופן הבא:  
השורה הראשונה בקובץ היא מספר יחיד המייצג את כמות הקודקודים ברשת.  
לאחר מכן כל שורה מייצגת פרטים עבור קודקוד מסוים בפורמט הבא:

node\_id neighbor1 weight send\_port listen\_port neighbor2 weight ...

המספר הראשון בשורה מייצג את מזהה הקודקוד. לאחר מכן מופיעים הפרטים על השכנים של אותו קודקוד, כאשר הפרט הראשון הוא המזהה של הקודקוד, השני הוא המשקולת על הקשת ביניהם, השלישי הוא הפורט עליו הקודקוד צריך לשלוח הודעות מהשכן שלו, והרביעי הוא הפורט ממנו הקודקוד מקבל את ההודעות מהשכן שלו. לדוגמה השורה הזו מהקובץ input1:

1 4 8.9 6060 13821 3 7.5 19068 6327

לקודקוד בעל המזהה 1 יש שני שכנים, שכן 4 שמשקולת הקשת ביניהם היא 8.9, ושכן 3 שמשקולת הקשת ביניהם היא 7.5. קודקוד 1 שולח הודעות לשכן 4 דרך פורט 6060 ומקבל ממנו הודעות דרך פורט 13821.

שימו לב כי מספרי הפורטים הם דו כיוונים, כלומר אם קודקוד 1 שולח הודעות לקודקוד 4 דרך פורט 6060, אזי קודקוד 4 מקבל את ההודעות מקודקוד 1 דרך פורט 6060. אין צורך להיכנס לפאניקה, כל הפרטים האלו קיימים גם בשורת הקלט הרלוונטית עבור קודקוד 4:

4 1 8.9 13821 6060 2 1.0 17757 28236 5 1.5 1603 24233 3 6.6 27781 1213

השורה stop מסמלת את סיום פרטי יצירת הגרף ועל ExManager והקודקודים להפסיק לקרוא את קובץ הטקסט בשורה זו. השורות הבאות יכילו עדכוני קשתות והוראות להרצה של אלגוריתם link state routing, שאינן רלוונטיות למחלקה ExManager. **אין להמשיך לקרוא את הקובץ לאחר הגעה לשורה stop.**

### טיפים של אלופים

1. התרגיל כולל חלקים של תקשורת באמצעות ממשקי שרת לקוח. עליכם יהיה להגדיר ממשקי שרת לקוח וכן להשתמש בתהליכים אותם לפעמים קשה יותר לדבג. מומלץ לוודא שכל המנגנונים הבסיסיים, לדוגמה יצירה של קודקוד והחזרה של קודקוד, עובדים לפני שמגיעים לממש את שלב התקשורת.
2. התרגיל כולל המון עבודה עם תהליכים, כל קודקוד רץ בתהליכון משלו וה"הקשבה" של קודקוד לכל אחד מהפורטים נעשית גם היא בתהליכון נפרד (יש לאפשר להודעות להגיע במקביל). מומלץ מאוד לחזור על החומר בנושא תכנות מקבילי שנלמד בקורס הנדסת תוכנה. בנוסף, שימו לב לקטעים הקריטים של הקוד והשתמשו במנעולים\ synchronized בצורה מושכלת.
3. הקפידו על כללי הנדסת תוכנה נכונה, השתמשו בתיעוד מתאים, נוטציות, הורשות, חלוקה לפונקציות ותבניות עיצוב במידת הצורך. הדבר יעזור לכם לכתוב ולבדוק את התרגיל בצורה נוחה ומהירה יותר.
4. התרכזו בהתחלה בשני קבצי הפלט הראשונים, הם מכילים גרפים קטנים שניתן לשרטט ביד וקל לעקוב אחריהם.
5. אתם מוזמנים לנסות וליצור קבצי קלט נוספים בעצמכם. זכרו כי פורטים 0 עד 1023 הינם פורטים שמורים לשימושי המחשב, אך אחריהם יש לכם עוד כ-64000 פורטים פנויים (:
6. אל תהססו להיעזר באינטרנט.