

## תרגיל בית מספר 5: מטלת סיכום

**נושא: עבודה עם גרפים בעזרת הספרייה הסטנדרטית STL**

**דדליין: יום ראשון, 29/01/2023, 23:59**

**הגשה אפשרית בזוגות**

### בהצלחה רבה!

#### תיאור התרגיל

תרגיל זה עוסק באפיון של רשת תחבורה באמצעות גרפים, ובמימושו אתם מוזמנים להפעיל את כל כלי התכנות שרכשתם במהלך הקורס. בתרגיל זה עליכם לעשות שימוש בספריית STL, ובנוסף עליכם גם לשלב מנגנון חריגות לטיפול בשגיאות.

#### חלק א': גרף

בחלק זה עליכם לתכנן ולממש גרף מכוון וממושקל, תחת דרישה למימוש של המרכיבים הבאים:

- הוספת/מחיקת קודקוד לגרף
- הוספת/מחיקת קשת מכוונת וממושקלת לגרף
- עדכון משקל של קשת נתונה בגרף
- שאילתא על משקל של קשת נתונה בגרף
- שאילתא על זהות השכנים הסמוכים של קודקוד נתון בגרף (הקודקודים אליהם ניתן להגיע במעבר קשת יחיד)
- מציאת קבוצת הקודקודים המקושרים לקודקוד נתון (כלומר, קבוצת הקודקודים אליהם ניתן להגיע במספר כלשהו של צעדים)
- שאילתא על המקורות הסמוכים אל קודקוד נתון בגרף (הקודקודים מהם ניתן להגיע אליו במעבר קשת יחיד)
- הדפסת הגרף. עבור כל קודקוד יש להדפיס את השכנים הסמוכים מופרדים ע"י רווח.

**אילוצים:** עליכם לממש את הגרף כמטריצה מעל טיפוס הווקטור הסטנדרטי (`std::vector`), כלומר, הייצוג הפנימי של הגרף יהיה באמצעות מטריצת שכנויות ממטית המכילה מידע על משקולות הקשתות. פרט לכך, אתם רשאים לממש את הגרף כרצונכם, וכן להוסיף פונקציות נוספות, כל עוד הפונקציונאליות המתוארת תישמר. חלק ב' של תרגיל זה יעשה שימוש בגרף לשם אפיון רשת תחבורה ובדיקת קישוריות בין נקודות ברשת.

#### חלק ב': אפיון רשת תחבורה בגרף מכוון וביצוע בדיקות קישוריות

משרד מהנדסי תחבורה בארץ-שום-מקום עוסק בתכנון קווים עירוניים בעיר הבירה Neverland City. לשם כך, המשרד זקוק למערכת ממוחשבת המכילה את קווי התחבורה הקיימים בעיר ואשר מסוגלת לקבוע את הקישוריות של נקודות שונות בעיר. עליכם לתכנן ולממש תכנית המקבלת כקלט קבצים המתארים את רשת התחבורה הקיימת, ולאחר אחסונה בזיכרון, מאזינה לערוץ קלט סטנדרטי מול המשתמש בטרמינל לשם קביעת הקישוריות של נקודות מסוימות ברשת. בשלב שלילת המידע, המשתמש יקליד נקודת ציון בעיר (שם של נקודה ברשת) ויקבל כפלט את הנקודות אליהן ניתן להגיע מנקודה זו באמצעות הרשת הקיימת; לאחר שלילה זו, התכנית תחזור למצב המתנה לקלט נוסף, עד אשר תוקלד פקודת יציאה.

#### קלט

התכנית תקבל כארגומנטים מספר בלתי מוגבל של שמות קבצי קלט, וקיימת אופציה למתן שם קובץ פלט. לאחר קומפילציה של התכנית, למשל לכדי קובץ הרצה בשם `traverseNeverlandCity`, כך ניתן יהיה להריץ אותה במקרה של עבודה דרך טרמינל:

```
>> ./traverseNeverlandCity <infile1> <infile2> <infile3> ... [-o] <outfile>
```

ללא שימוש באופציה לציון שם קובץ הפלט (שימוש בדגל `-o`), ברירת המחדל עבור הפלט תהיה קובץ בשם `output.dat`.

# תכנות בשפת C++, סתיו 2014-15

שורת קלט חוקית בכל אחד מן הקבצים תכלול את נקודת המוצא, את נקודת היעד, ואת זמן הנסיעה של קו מסוים, מופרדים באמצעות טאב:

```
<source_node> <TAB> <target_node> <TAB> <hop_time>
```

משמעות של שורה כזו היא שקיים קו תחבורה ישיר בין המוצא לבין היעד, עבורו נתון זמן הנסיעה ביחידות של דקה; קו ישיר כזה יתורגם לקשת מכוונת בגרף שמשקלה הוא זמן הנסיעה. הטענה כי קיים קו בכיוון ההפוך בין היעד לבין המוצא איננה נכונה אלא אם כן מופיעה שורת קלט נוספת המציינת זאת. להלן דוגמא מפורשת עבור שורת קלט חוקית:

CSWestzijde Prinsengracht 10

- שם נקודה ברשת הינו מחרוזת אלפביתית שאינה עולה באורכה על 16 תווים ואיננה כוללת את תו הרווח.
  - המחרוזת exit איננה שם חוקי של נקודה ברשת
- זמן נסיעה חוקי בין שתי נקודות בקשת הוא בהכרח מספר שלם חיובי.
- בהינתן מידע סותר על קשת שכבר קיימת בגרף (תסריט של קשת שהתווספה לגרף על-סמך נתונים של קובץ אחד, ומופיעה שוב בהמשך קריאת הנתונים מקובץ נוסף עם משקולת אחרת), משקולת הקשת תעודכן ע"פ המשקל הנמוך ביותר.
- במקרה של קלט לא חוקי (למשל, שגיאה בפתיחת אחד הקבצים, טקסט בלתי-חוקי, פורמט שגוי, וכיוצא באלה), מנגנון חריגות (Exceptions), שעליכם לממש, יופעל. בעקבותיו התכנית תסיים את ריצתה בשלב זה ותדפיס הודעת שגיאה מתאימה לערוץ השגיאות הסטנדרטי.

## פלט

בסיום בניית הגרף, על-סמך קבצי הקלט השונים, על התכנית לרשום לקובץ הפלט את ייצוג הגרף באמצעות פונקציית ההדפסה שמימשתם בחלק א'.

## אינטראקציה מול המשתמש

לאחר קריאת הנתונים, בניית הגרף המתאר אותם והדפסתו אל קובץ הפלט, התכנית תעבור למצב המתנה לטקסט של המשתמש בערוץ הקלט הסטנדרטי.

- במצב זה, על המשתמש בתכנית להקליד שם נקודה ברשת העירונית, אותה נציין כ- <source\_node>, ולאחר הקלדת Enter, התכנית תדפיס לפלט הסטנדרטי את הנקודות ברשת אליהן ניתן להגיע (מ) מנקודה זו באמצעות רשת התחבורה הקיימת בפורמט הבא:

```
<source_node> <TAB> <node_1> <TAB> <node_2> <TAB> . . .
```

- אם לא קיימים ברשת נקודות המקושרות לנקודה <source\_node>, התכנית תדפיס אל הפלט הסטנדרטי את ההודעה הבאה:

```
<source_node> <SPACE> : <SPACE> no outbound travel
```

- אם לא קיימת ברשת נקודה כפי שצוינה ע"י המשתמש כ- <source\_node>, התכנית תדפיס אל הפלט הסטנדרטי את ההודעה הבאה:

```
<source_node> <SPACE> does not exist in the current network <\n>
```

```
USAGE: <node> or 'exit' to terminate
```

בכל מקרה, לאחר מכן התכנית תמתין לקלט נוסף מן המשתמש. הקלדת הפקודה exit תסיים את התכנית.

# תכנות בשפת C++, סתיו 2014-15

## דגשים

- יש לבדוק תקינות קלטים; במקרה של אי-תקינות, יש להפעיל מנגנון חריגות ולהציג הודעות שגיאה מתאימות.
- כאמור, בתרגיל זה עליכם להשתמש בספריית STL בכלל, ובמבנה `std::vector` בפרט (במימוש הגרף).
- עליכם לוודא כי התכנית עוברת קומפילציית g++ התואמת את הקומפיילר שעל שרת המכללה ללא כל שגיאות או אזהרות כלשהן, ורצה בהצלחה.
- עליכם לתעד את הקוד באמצעות הערות המתארות בקצרה את המחלקות והפונקציות השונות.

מנגנון חריגות (Exceptions), שעליכם לממש, יופעל. בעקבותיו התכנית תסיים את ריצתה בשלב זה ותדפיס הודעת שגיאה מתאימה לערוץ השגיאות הסטנדרטי:

1. באם לא קיים קובץ שניתן לתכנית באמצעות המחזור `<filename>`, או פתיחתו לקריאה/כתיבה נכשלת, על הודעת השגיאה לציין את שם הקובץ באופן הבא (במקרה ששני הקבצים נכשלים, יש להדפיס רק את כישלון הקלט):  
`ERROR: <filename> does not exist or cannot be opened\n`
2. באם קיימת שגיאה בהגדרת הקשת של קובץ הקלט `<input>`, על הודעת השגיאה לציין זאת באופן הבא:  
`ERROR: node definition in <input> is invalid\n`

- בהנחה שהקלט של המשתמש חוקי, התכנית תתחיל לסמלץ את מעברי הזמן עד שמספר המעברים המתבקש יחלוף.

## דגשים

- יש לבדוק תקינות קלטים; במקרה של אי-תקינות, יש להפעיל מנגנון חריגות ולהציג הודעות שגיאה מתאימות.
- כאמור, בתרגיל זה ניתן להשתמש בספריית STL במלואה בחלק ב' בלבד (בחלק א' ניתן לעבוד עם `container` שלה).
- עליכם לתעד את הקוד באמצעות הערות המתארות בקצרה את המחלקות והפונקציות השונות.
- עליכם לוודא כי התכנית עוברת קומפילציית g++ התואמת את הקומפיילר שעל שרת המכללה ללא כל שגיאות או אזהרות כלשהן, ורצה בהצלחה.
- יש להריץ את הבודק האוטומטי על שרת החוג בטרם ההגשה בכדי לוודא תאימות ונכונות של ההגשה: התחברו לשרת החוג והריצו `hwcheck` על הארכיב שלכם, או לחילופין העלו את הארכיב תוך שימוש בפרוטוקול HTML בקישור <https://csweb.telhai.ac.il>

## הגשה

- עליכם להגיש במערכת Moodle קובץ ארכיב מטיפוס zip בלבד, ששמו כולל את קוד הקורס ('44'), שם התרגיל ('ex5') ותעודת הזהות של הסטודנט/ית המגיש/ה, מופרדים בקו תחתי באחד מן הפורמטים הבא:  
`44_ex5_studID1_studID2.zip` או `44_ex5_studID.zip` במקרה של הגשת צמד.
- על ארכיב zip זה להכיל את כל קבצי המקור (ממשק/מימוש) הנדרשים לקומפילציה, והוא רשאי להכיל תיעוד טקסטואלי; מבחינת טיפוס קבצים, עליו לכלול רק קבצים עם סיומות `*.cpp` `*.h` `*.txt`.
- לדוגמה: על סטודנט שמספר הזיהוי שלו הינו 012345678 להגיש ארכיב בשם `44_ex5_012345678.zip` הכולל את כל קבצי המקור של הפרוייקט, ללא תיקיות כלשהן, ורשאי להכיל קובץ טקסטואלי לתיעוד.

אי-הקפדה על ההנחיות, כולל פורמט ההגשה הדיגיטלי, תגרור הורדה בציון התרגיל.  
לא תתקבלנה הגשות באיחור!