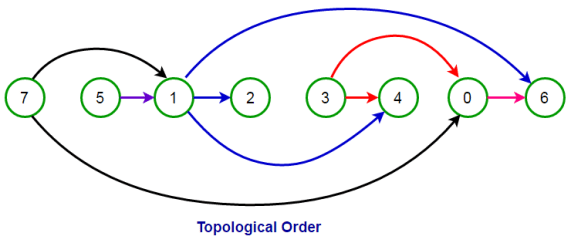
בס"ד

# תרגיל 5 – גרפים

מטרת תרגיל זה: בניית תת מערכת שתאפשר לקבוע סדר של מטלות שיש צורך לבצע לטיפול במצבים מסוימים במוקד הרפואי.

נניח שקיימות סדרה של מטלות שיש לבצע במצב מסוים, ועבור כל מטלה, מוגדרת רשימת מטלות שצריכות להסתיים לפני הביצוע שלה.

לצורך כך עליכם להגדיר מבנה של גרף מכוון של מטלות, כאשר כל מטלה מוגדרת כקדקוד וכל סדר בין מטלות מוגדר כקשת. לדוגמא, אם מטלה A יכולה להתבצע רק אחרי שמטלות B,C מסתיימות, תוגדר קשת מקודקוד B לקדקוד A, וקשת נוספת מקודקוד C לקדקוד A. (כפי שלמדנו בשיעור).



**בתרגיל זה עליכם להגדיר את הגרף, לקלוט את רשימת המטלות כפי שתוארה למעלה, ולהדפיס הצעה לסדר של ביצוע המטלות, באופן שכל מטלה תתבצע רק אחרי סיום ביצוע המטלות שצריכות להסתיים לפניה.**

**שלב א': הגדרת הגרף**

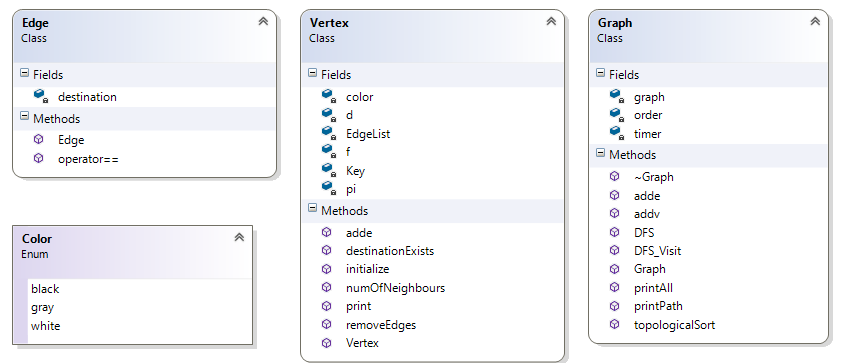
**יש להגדיר מחלקה עבור גרף מכוון כאשר כל קדקוד בה מכיל מפתח מסוג string**

* הגדירו מחלקה עבור **קדקוד**, שתכיל את המפתח שלו (מסוג string), שאמור להיות שם של מטלה, ורשימה מקושרת של **קשתות** ופרטים נוספים לפי צורך.

הגדירו מחלקה עבור **קשת**, שתכיל מצביע לקדקוד היעד

הגדירו מחלקה מסוג **גרף מכוון** שתשמור בתוכה את כל הקדקודים בגרף. (אפשר לשמור אותם בצורה של טבלת גיבוב של מחרוזות, שהמפתח שלהן הוא שם המטלה, ותוכן כל כניסה הוא המצביע לקדקוד. אפשרות נוספת היא להגדיר map של מפתח ומצביע לקדקוד, כך שיתאפשר להגיע לקדקוד במהירות בהינתן המפתח שלו).

**ClassDiagram:**



המימוש של הגרף יהיה באמצעות רשימות שכנים (לכל קדקוד, רשימה של קשתות לכל שכניו).

* בנאי (מאתחל את הגרף כגרף ריק) והורס.
* הוספת קדקוד לגרף.
* הוספת קשת לגרף (יש לוודא שלא מוסיפים קשת שכבר קיימת).
* בהינתן קדקוד, החזרת רשימה מקושרת של הקשתות היוצאות ממנו.
* הדפסת הגרף: לכל קדקוד בגרף להדפיס את כל הקשתות היוצאות ממנו.
* אלגוריתם של DFS על הגרף כולו, שיחזיר רשימה מקושרת ממוינת בסדר יורד של זמני סיום.

(כפי שהוזכר בתחילת התרגיל, השימוש שלנו בגרף יהיה באופן הבא: כל מטלה מיוצגת על ידי קדקוד, והקשתות מייצגות סדר בין מטלות).

|  |
| --- |
| **הערה חשובה**: כדי במצב שבו יש הדדיות (קדקוד משתמש בקשת וקשת משתמשת בקדקוד) צריך להצהיר על המחלקה של קשת לפני ההגדרה של המחלקה של קדקוד: כמו שהיה בתרגיל 1. |
| Example:  class Vertex; //הצהרה  class Edge  {  Vertex\* destination;  public:  friend class Vertex;  ……  };  class Vertex  {  String Key;  list<Edge \*>EdgeList;  double d;  double f;  Vertex \*pi;  public:  Vertex();}; |

**מומלץ לבדוק את המימוש לפני שעוברים לשלב ב'**

**שלב ב': תכנית ראשית שלב זה ( שלב ב בלבד ) מומש והועלה למודל**

|  |
| --- |
| יש לאפשר לבצע שוב ושוב את הפעולות הבאות:   * קליטת מטלה חדשה ובניית קדקוד מתאים. * קליטת סדר בין שתי מטלות (מטלה שיכולה להתבצע רק עם סיום מטלה אחרת) ובניית קשת מתאימה. * הדפסת הגרף כולו. * הדפסת סדר מומלץ עבור המטלות, באמצעות ביצוע מיון טופולוגי של המטלות והדפסת הרשימה המקושרת שהתקבלה. * יציאה: מחיקת כל מבני הנתונים שהוקצו. |

**בהצלחה רבה!**