

מטלה 1 תכנות מערכות 2

המחלקה Graph:

במחלקה זו ישנם 3 שדות:

מספר הקודקודים בגרף- numOfVertices, אותו חישבתי על ידי אורך המטריצה (כלומר אם המטריצה היא 5×5 אז מספר הקודקודים הוא 5), בנוסף קיימת בדיקה האם הגרף ריבועי או לא, במידה והוא לא נחזיר שגיאה

מספר הצלעות בגרף- numOfEdges, מחושב על ידי מעבר על מטריצה וכל מקום בו אין 0 מזהה בתור צלע מ- i אל j

מטריצת שכנויות המייצגת את הגרף- vec, אותו אנחנו מקבלים מהמשתמש

פונקציה loadGraph- מקבלת מטריצה המייצגת את מטריצת השכנויות שלנו ומבצעת השמה לשדה vec, בנוסף סופרת את מספר הצלעות שיש שבגרף ומבצעת השמה בשדה NumOfEdges

פונקציה printGraph- מדפיסה את מספר הקודקודים והצלעות שיש בגרף

המחלקה Algorithms:

מכילה 5 פונקציות עיקריות על גרפים, נעבור אחד אחד עליהם-

isConnected- בודקת האם הגרף קשיר חזק, כלומר בודקת האם קיים מסלול מכל קודקוד אל כל קודקוד אחר, אם כן מחזירה TRUE אחרת מחזירה FALSE. בעזרת פונקציות העזר isPathHelp ו isPath עוברים על כל זוגות הקודקודים האפשריים ובודקים האם קיים מסלול ביניהם, זה נעשה על ידי הליכה על הגרף מקודקוד src אל קודקוד des באופן רקורסיבי כאשר בכל צעד רקורסיבי הולכים על קודקוד אחד בלבד ומנסים להגיע אל des, במידה והגענו אליו סימן שקיים מסלול מ src ל des ונחזיר TRUE.

isContainsCycle- עוברים קודקוד קודקוד ומנסים למצוא מכל אחד מהם מעגל, זה נעשה על ידי שמירה של מערך בוליאני שמסמן האם במהלך ההליכה שלנו על הגרף כבר היינו על כל קודקוד, אם הגענו אל קודקוד שכבר היינו עליו סימן שיש מעגל, ואז נשחרר את המעגל הזה אל תוך מערך, נדפיס אותו ונחזיר TRUE. הפונקציה הראשית נעזרת בפונקציית העזר hasCycleDFS כאשר בכל פעם הפונקציה הראשית שולחת קודקוד אחר (בודקים אם יש מעגל מכל קודקוד) ופונקציית העזר מחפשת עבורו מעגל, במידה והיא לא מוצאת נחזיר FALSE.

shortestPath- מקבלת גרף, קודקוד התחלה וקודקוד סיום ומדפיסה את המסלול הקצר ביותר (או הקל ביותר אם מדובר על גרף עם משקלים) מקודקוד ההתחלה לקודקוד הסיום. זה נעשה על ידי מעבר של אלגוריתם בלמן-פורד על הגרף, ראשית מאתחלים מערך שיכיל את משקל המסלול הקל ביותר מקודקוד ההתחלה אל כל קודקוד אחר (בהתחלה כולם מאותחלים ל- max integer), לאחר מכן מבצעים n-1 פעמים מעבר על כל הצלעות ומבצעים את פעולת relax עליהם כך פעם מחדש. לאחר מכן, נבצע מעבר אחרון על כל הצלעות ואם

עדיין אפשר לעשות relax זה אומר שקיים מעגל שלילי בגרף ונחזיר שגיאה. במידה ועברנו את זה בהצלחה אזי במערך המקורי יהיה לנו את המרחק הקל ביותר מקודקוד ההתחלה לכל קודקוד אחר בגרף, נסתכל על המקום של קודקוד הסיום שקיבלנו במערך, אם הוא max integer אזי נחזיר 1- מכיוון שאין דרך להגיע מקודקוד ההתחלה אליו, במידה והוא לא אזי נשחר את הצעדים מקודקוד הסיום אל קודקוד ההתחלה, זה יהיה המסלול ונחזיר אותו.

isBipartite - בודק האם הגרף הוא דו צדדי או לא. מתבצע על ידי ניסיון צביעת הגרף לשני צבעים, מתחילים מהקודקוד הראשון (הקודקוד שמסומן בתור 0) וצובעים אותו בצבע כלשהוא, לאחר מכן עוברים על כל השכנים שלו ובודקים 3 דברים: 1) אם השכן אינו צבוע אזי נצבע אותו בצבע הנגדי של הקודקוד המקורי, 2) אם השכן צבוע בצבע של הקודקוד המקורי אזי הגרף אינו חד צדדי ונחזיר FALSE, 3) אם השכן צבוע בצבע הנגדי של הקודקוד המקורי אזי נוסיף אותו לקבוצה הנגדית של הקודקוד המקורי. וכך נעבור על כל הקודקודים בגרף ונחלק אותם ל2 קבוצות, אם המעבר עבר חלק ולא נתקלנו בבעיה אזי נחזיר את החלוקה ל2 קבוצות, אם לא אזי נחזיר שהגרף אינו חד צדדי.

negativeCycle - בודק האם קיים מעגל שלילי בגרף. מתבצע על ידי הרצת אלגוריתם בלמן פורד על הגרף, כמו שביצענו בפונקציית shortest path, לאחר ביצוע n-1 פעמים (n זה מספר הצלעות) את הפעולה relax אזי נרוץ פעם אחת אחרונה על כל הצלעות, אם ניתן לצבע שוב relax על צלע כלשהיא אזי זה אומר שיש מעגל שלילי בגרף, אם אין פעולת relax אזי נחזיר שאין מעגל שלילי בגרף.