שאלה 1:

* רעיון האלגוריתם:
  + שימוש באלגוריתם הזיווג היציב שמתואר בפרק 1 בספר כאשר נבצע ההקבלה של הספינות לגברים, והנמלים לנשים.
  + נתאים כל ספינה לנמל, כאשר כל התאמה מייצגת ספינה שהגיע לנמל ונשארת שם עד סוף החודש לעבודות תחזוקה.
  + נשים לב שמספר הנמלים ומספר הספינות זהה (n) כדרוש באלג' הזיווג היציב, ואכן לא יתכנו 2 ספינות באותו הנמל (במקביל לפוליגמיה בזיווג היציב).
  + נסמן ב S את קבוצת הספינות וב-P את קבוצת הנמלים.
* הגדרת ההעדפות:
  + העדפות של ספינה:
    - לכל ספינה נגדיר את העדפה של ספינה לנמל על פני נמל אם"ם נמל נמצא בלוח הזמנים של לפני נמל , כאשר .
    - כך שהנמל הראשון אליו מגיעה הספינה נמצאת בעדיפות גבוהה יותר שלה, הנמל השני בעדיפות השנייה וכן הלאה.
  + העדפות של נמל:
    - לכל נמל נגדיר את העדפה של נמל לספינה על פני ספינה אם ספינה מגיעה לנמל לפי לוח הזמנים המקורי אחרי ספינה , כאשר .
    - כך שהספינה שצריכה להגיע אחרונה לנמל p - תהיה בעדיפות הראשונה שלו, הספינה שצריכה להגיע לפניה תהיה בעדיפות השנייה וכן הלאה.
* תיאור האלגוריתם:
  + בניית רשימת ההעדפות של הספינות מתוך הלו"ז המקורי.
    - לכל ספינה , נסרוק את הלו"ז המקורי שלה מ1 עד m , כאשר כל נמל שמופיע בלו"ז יכנס לסוף הרשימה.
  + בניית רשימת ההעדפות של הנמלים מתוך הלו"ז המקורי.
    - נעבור בלולאה על כל m הימים של החודש בסדר יורד (החל מm עד 1) d -
      * נעבור על כל הספינות בסדר יורד (n עד 1) s -
        + ונבדוק האם ברשימת ההעדפות המקורית הספינה s עוגנת בנמל p כלשהו ביום d.

אם כן – נכניס לסוף רשימת ההעדפות של p את s.

אחרת – נמשיך לאיטרציה הבאה של הלולאה.

* + ריצת אלגו' הזיווג היציב:
    - בתחילת הריצה – הספינות והנמלים אינם משוייכים זה לזה.
    - כל עוד קיימת ספינה s פנויה שעדיין לא הציעה נישואין לנמל כלשהו.
      * ספינה זו תציע נישואין לנמל הראשון ברשימת ההעדפות שלה, שלא הציע לו בעבר, נסמן אותו ב p
        + אם p לא מאורס – הוא יסכים להצעה, וs,p יתארסו.
        + אם p מאורס לספינה אחרת אך s גבוהה יותר ברשימת ההעדפות שלו 🡨 האירוסין עם s’ יתבטלו וs,p יתארסו.
        + אחרת (p מאורס ל s’ וגם s נמוכה ברשימת ההעדפות של p) – לא יתבצעו אירוסין וs תעבור בלולאה לנמל הבא ברשימת ההעדפות שלה.
    - בסיום הלולאה – כל קשרי האירוסין יהפכו לנישואין וסיימנו את תהליך השידוך.
  + אלג' הזיווג היציב ימומש כמתואר בסעיף 2.3 בספר כדי לוודא ריצה בזמן אופטימלי.
* נכונות האלגוריתם:
  + כדי להתאים את נכונות האלגו' לספר – נראה כי מתקיימים התנאים הדרושים:
    - ברשימת ההעדפה של כל נמל מופיעות כלל הספינות בדיוק פעם אחת 🡨 מתקיים מתוך הנתון שכל ספינה עוגנת בכל נמל בדיוק פעם אחת.
    - באופן דומה – ברשימת ההעדפות של כל ספינה מופיעים כלל הנמלים.
    - בסיום הריצה – קשר הנישואין הוא מחייב ולפיו תישאר הספינה בנמל עד לסוף החודש לעבודות התחזוקה.
  + נוכיח כי מתקיים האילוץ בו 2 ספינות לא יהיו יחדיו באותו הנמל.  
    כלומר שאם s,p נשואים, כאשר s מגיעה לp ביום d – לא תגיע אף ספינה לנמל p בימים:   
    - נניח בשלילה שs,p נשואים כאשר s מגיעה ל p ביום d, וספינה s’ מגיעה ביום t כאשר
    - נובע מכך שהנמל p היה ברשימת ההעדפות של s’ במקום גבוה יותר מאשר המיקום של p ברשימה של s.
    - ועל כן – כאשר יצטרך הנמל p לבחור ספינה – הוא יעדיף את s’ על פני s
    - וזו סתירה לכך ש s,p נשואים.
    - מש"ל
* סיבוכיות האלגוריתם:
  + חישוב רשימת ההעדפות של הספינות מתבצעת עבור כל אחת מ n הספינות באמצעות מעבר על כל אחד מm הימים. 🡨
  + חישוב רשימת ההעדפות של הנמלים מתבצעת עבור כל אחד מm הימים בחודש, עבורו סורקים את כל n הספינות. 🡨
  + זמן הריצה של אלגו' הזיווג היציב הוא לפי עמוד 52 בספר.
  + לאחר סכימה נקבל:
  + ומכיון שנתון כי נקבל:

שאלה 2:

* רעיון האלגוריתם:
  + כיוון הגרף כאשר לכל צומת יש דרגת כניסה גדולה מ-0 יתקיים כאשר בגרף יש לפחות מעגל אחד. נתחיל מצומת כלהיא ונכוון באמצעות סריקת BFS עם שינויים קלים.
* תיאור האלגוריתם:
  + נבחר *קודקוד כלשהו ונריץ עליו סריקת DFS , נסמן את העץ שקיבלנו ב T.*
    - *אם כל הקשתות ב T הן קשתות עץ (ללא קשתות חוזרות) – הרי שלא קיים מעגל 🡨*
      * *נחזיר False מאלגוריתם.*
    - *אחרת, קיימת קשת חוזרת . נכוון אותה כ.*
      * *ניגש לכל הקשתות הלא-מכוונות שיוצאות מv ונכוון אותן כך שיצאו מv – "החוצה". (ביצוע BFS כך שלכל רמה i תשלח קשת מכוונת לרמה i+1)*
        + *בצורה רקורסיבית עבור כל קשת לא מכוונת – נכוון אותן "כלפי חוץ"*
        + *בכל קביעת כיוון של קשת – נדפיס את הכיוון שנקבע*
        + *בסוף הריצה (כאשר נגיע חזרה ל v) 🡨 נחזיר True.*
* נכונות האלגוריתם:
  + T הוא תוצאה של סריקת DFS ולכן יש לו 2 סוגי קשתות: קשתות עץ וקשתות חוזרות.
  + אם כל הקשתות הן קשתות עץ אזי * , ולכן – G הוא עץ.*
    - *נסמן ולכן לפי משפט 3.1 , .*
    - *נניח בשלילה שניתן להגדיר גרף מכוון -*
    - *אזי מתקיים ,* ***אבל*** *אם סך דרגות הכניסה הוא* **n-1** *וכזכור מספר הצמתים הוא* **n***אז בהכרח קיים קודקוד שדרגת הכניסה שלו היא 0* 
      * *נחזיר False*
  + אחרת – *קיימת לפחות קשת חוזרת אחת, נסמנה . ונכוון אותה כ.* 
    - *לv בהכרח יש קשתות-יוצאות (אחרת לא היתה קשת חוזרת)*
    - *הכוונת הקשתות מתבצעת בצורה המזכירה BFS לפי שכבות, כאשר v הוא השורש (כל הקשתות של צמתים ברמה מצביעים על צמתים ברמה )*
    - *בצורה זו אכפנו לכל צומת - דרגת כניסה בגודל 1 לפחות.*
* סיבוכיות האלגוריתם
  + שימוש ב DFS לסריקת הגרף G ~
  + שימוש בוריאציה של BFS לסריקה וכיוונון הצמתים שאינם מכוונים ~
  + סה"כ -

שאלה 3:

* רעיון האלגוריתם:
  + נמיר את הפסוקית הלוגית לקשרי גרירה ונכניס אותה לגרף מכוון.
  + נבצע השמות על רכיבי קשירות נפרדים תוך ווידוא כי לא קיבלנו סתירה.
* תיאור האלגוריתם:
  + נבנה גרף מכוון כך:
    - לכל ליטרל x ניצור את הקודקודים: .
    - לכל פסוקית מהצורה , ניצור את הקשתות:
      * לפי השקילות הלוגית, עבור 2 פסוקים p,q :
  + *כל עוד קיים קודקוד שלא עבר השמה:*
    - *נבחר קודקוד כזה ונריץ סריקת DFS ממנו, נסמנה K.*
      * *אם לא נמצא ב, נבצע את ההשמה: T לכל קודקוד בקבוצה שנמצאה, ו-F לכל הקודקודים הנגדיים שלהם.*
      * *אחרת, נריץ סריקת DFS על , נסמנה R.*
        + *אם לא נמצא ב, נבצע את ההשמה: נגדיר T לכל קודקוד בקבוצה שנמצאה, ו-F לכל הקודקודים הנגדיים שלהם.*
        + *אחרת, נחזיר False.*
  + *בסיום השמה מוצלחת (בה לא קיבלנו סתירות) 🡨 נחזיר True.*
* נכונות האלגוריתם:
  + בכל איטרציה על צומת v שעדיין אין לו השמה – אנחנו מבצעים DFS על כדי למצוא כל הצמתים ברכיב הקשירות שלו, אם נמצא כבן של v , אזי נקבל סתירה במסלול הפסוקי.
  + **אבל**, **המסלול ההפוך עשוי להתקיים**, כלומר אם קיים מסלול מ שאיננו מגיע לv ההשמה שלו לא בהכרח מובילה לסתירה. ולכן במידה והחלק הראשון הופרך – נריץ DFS על ונבדוק האם v הוא בן של :
    - אם כן – אזי הצמתים נגישים הדדית 🡨 בכל השמה נקבל סתירה 🡨 נחזיר False.
    - אם לא – אזי נבצע השמה לכלל הבנים של ונמשיך באיטרציה.
  + אם הגענו למצב בו לכל צומת יש השמה ולא קיבלנו סתירות – נחזיר True.
* סיבוכיות האלגוריתם
  + בניית הגרף G ~
  + ריצה בלולאה על כל הצמתים שעוד לא בוצעו להם השמות:
    - במקרה הגרוע ( בן של v) נבצע DFS פעמיים על כל איטרציה בלולאה הראשית.
    - נסמן בk את מספר הקודקודים עליהן נרוץ בלולאה הראשית , כאשר .
      * מתבצעות 2k השמות לליטרלים
  + עדיין מספר הקודקודים שאנחנו מבצעים להם השמה הוא קבוע, לא ייתכן שנבצע השמה פעמיים. ולכן בכל מקרה נישאר בחלק זה ב
  + סה"כ -

שאלה 4:

* רעיון האלגוריתם:
  + בדומה לבניית אוטומט-סופי-דטרמיניסטי - נחלק את המסלול ל-5 שלבים.
  + כאשר  *(X קבוצת קודקודים מתוך V* ***שאינם******קודקודים******מועדפים****,* ***ללא*** *s ו-t).*
  + *נוכיח בצורה קונסטרוקטיבית כיצד לבנות גרף מכוון חדש G’ שמורכב מ-5 השלבים, לבסוף נמצא את המסלול המינימלי באמצעות סריקת BFS (אם קיים).*
* תיאור האלגוריתם:
  + *נגדיר*
  + *נבנה מ גרף מכוון חדש באופן הבא:*
    - *לכל קודקוד מועדף , .*
    - *את נעתיק לגרף החדש.*
    - *לכל קודקוד ניצור בגרף החדש את .*
      * *לכל קשת ,*
      * *לכל קשת ,*
      * *לכל קשת ,*
      * *לכל קשת ,*
      * *לכל קשת ,*
      * *לכל קשת ,*
      * *לכל קשת ,*
    - *כעת נריץ סריקת BFS על הגרף החל מ s. עבור ההופעה הראשונה של t בסריקה, המסלול מs עד t הוא המסלול הקצר ביותר שמקיים את תנאי השאלה. אם לא נמצא t בסריקה – אזי לא קיים מסלול כזה.*
* נכונות האלגוריתם:
  + אילוץ 1 -
    - באמצעות תיוג הקודקודים מהקבוצה U והגדרת קשתות בהתאם – אנחנו אוכפים למעשה את המעברים בין השלבים השונים, דילוג על שלבים 2,4 לא.   
      (בדומה להגדרה פורמלית של אוטומט דטרמיניסטי).
    - הסבר המעברים לעיל (בהתאמה):
      * *מעבר מקודקוד התחלתי s אל איבר אחר בשלב 1*
      * *מעבר מקודקוד התחלתי s ישירות לשלב 2*
      * *מעברים משלב 1 ל2 ומשלב 3 ל4*
      * *מעבר משלב 2 ל3 ומשלב 4 ל5*
      * *מעבר משלב 2 ל4*
      * *מעבר משלב 4 לקודקוד הסופי t*
      * *מעבר משלב 5 לקודקוד הסופי t*
    - *כל מסלול בG’ המסתיים בשלב 5 מקיים את תנאי השאלה. (מתקבל עבור האוטומט)*
  + *אילוץ 2 – המסלול הקצר ביותר*
    - *שימוש בסריקת BFS מבטיח קבלה של המסלול הקצר ביותר , מכיוון שהסריקה מתבצעת בשלבים כאשר קודקוד שהופיע בשלב מוקדם – לא יופיע בשלב מאוחר יותר.*
  + *אילוץ 3 – המסלול שמצאנו קיים ב G*
    - *ע"י הסרה של התיוגים של הקודקודים והסימונים מהקשתות (במקביל להומומורפיזם באוטומט) – אנחנו מקבלים מסלול תקין בG.*
* סיבוכיות האלגוריתם
  + בניית הגרף G’ מתבצעת ע"י מעבר על ל הקודקודים והקשתות של G פעם אחת ולכן ~
  + סריקת BFS מעל G’ ~
  + סה"כ -