

9. בעיות מסכמות

פרק זה הינו הפרק האחרון בחומר הלימוד. מטרתו להציג בעיות סיכום ופתרונותיהן, תוך הדגשת תהליך הפתרון, אשר כולל שני מרכיבים מרכזיים – ניתוח בעיה המוביל לבחירת מבנה נתונים מתאים, ובחירת תבניות לביצוע החישובים הנחוצים במבנה הנתונים הנבחר. מבנה הנתונים יכול לכלול מרכיב אחד, מספר מרכיבים זהים, או מספר מרכיבים שונים. למשל, יתכן שבפתרון יהיה שימוש ברשימה אחת בלבד, או שימוש בכמה רשימות, או שימוש ברשימה ותור, וכן הלאה.

בחירת מבנה הנתונים המתאים נעשית על-פי מאפייני הבעיה, אשר נגזרים מאופי הנתונים ומאופי הפעולות הנדרשות עליהם. הנקודות הבאות מציגות שיקולים מנחים בבחירת מבנה נתונים מתאים.

- אם נתוני הבעיה הם סדרת ערכים שגודלה מוגבל וידוע מראש, ויש לשמור אותה ולאחזר ממנה שוב-ושוב מידע, מאיברים המצויים במקומות שונים בסדרה, אזי מתאים להשתמש במבנה הנתונים מערך לשמירת הסדרה וביצוע פעולות האחזור.
- אם יש לאפשר קליטת נתונים, מחיקת נתונים, ואחזור נתונים שונים, ללא מגבלה מוגדרת של כמותם, וללא קשר לסדר קליטתם, מתאים להשתמש ברשימה.
- אם עיבוד הנתונים הנחוץ מאופיין בסדר מסוג LIFO, פשוט או מורכב (למשל, מחזוריות של עיבודים מסוג LIFO), מתאים להשתמש במחסנית.
- אם עיבוד הנתונים הנחוץ מאופיין בסדר מסוג FIFO, פשוט או מורכב, מתאים להשתמש בתור.
- אם אופי הנתונים מבטא סדר היררכי, לא לינארי, וכך גם הפעולות על הנתונים, אזי עץ בינארי עשוי להתאים לשמירת הנתונים ועיבודם.

השיקולים בבחירת מבנה הנתונים נגזרים מן הנקודות המפורטות לעיל, תוך שאיפה למבנה נתונים פשוט ככל האפשר, שהפעולות עמו קריאות, "נוחות", ובעלות סיבוכיות נמוכה ככל האפשר. כאמור, לא-פעם יש צורך בשילוב מספר מרכיבים במבנה הנתונים. אלה נגזרים מהתייחסות לשילוב של הנקודות לעיל. כמו כן, לא-פעם יתכן שישנו יותר מפתרון (מבנה נתונים) אחד מתאים. עבור בעיה נתונה ייתכנו לפעמים מספר פתרונות חלופיים, אשר בכל אחד מהם שימוש במבנה נתונים אחר.

לאחר שנבחר מבנה נתונים מתאים יש להשתמש בתבניות מתאימות כדי לבצע את החישובים הנדרשים. בפרקים הקודמים הצגנו את התבניות הבסיסיות לשימוש בכל סוג של מבנה נתונים. מתאים וכדאי להיעזר בתבניות אלו בעת ניסוח אופן הביצוע של פעולות. לפעמים ניתן להשתמש בתבנית ככתבה, לפעמים יש לשנותה במקצת, ולפעמים יש להרכיב מספר תבניות יחד. לא-פעם כדאי גם לשלב תבניות שהצגנו בחומר לימוד קודם – של "יסודות מדעי המחשב" – שהן אבני בניין בסיסיות לכל פתרון אלגוריתמי. כמו כן, לפעמים מתאים לבחור בפתרון אלגוריתמי רקורסיבי, ולשם כך, כדאי להיעזר בתבניות של רקורסיה שהוצגו בתחילת חומר הלימוד. אנו מאמינים שהיכרות עם תבניות ומיומנות בשימוש בהן מסייעת באופן משמעותי וממקדת את הכתיבה בעת ניסוח הפסאודו-קוד והקוד של פעולות.

כדי להדגיש ולכוון לשני המרכיבים המרכזיים שהוזכרו לעיל בתהליך הפתרון: בחירת מבנה נתונים מתאים ושימוש בתבניות המתאימות לו, אנו מציגים בפרק סעיפי שאלות הפונים למרכיבים אלה. לדעתנו, מומלץ לכוון פותר בעיה בעיצוב תכנה לפי הסעיפים הבאים:

- א. תאר מאפיינים משמעותיים של הבעיה שעל-פיהם תבחר מבנה נתונים מתאים.
- ב. מהו מבנה הנתונים בו תשתמש לפתרון הבעיה?
- ג. מהן התבניות בהן תשתמש? וכיצד תשלב אותן ביחד?
- ד. נסח פסאודו-קוד לפתרון הבעיה, על-פי מבנה הנתונים והתבניות שבחרת.
- ה. מהי סיבוכיות הפתרון?
- ו. יתכן שנראית לך דרך נוספת לפתרון, תוך שימוש במבנה נתונים שונה (אולי פתרון זה יעיל פחות, אך עדיין רלוונטי). אם כן, ענה עבור דרך נוספת זו על הסעיפים ב-ה.

בפרקים קודמים (בפרט בפרקים אודות מחסנית ותור) הצגנו סעיפים כאלה או חלק מהם בבעיות המסכמות של פרק. בפרק זה אנו חוזרים, מדגישים ומכוונים לסעיפים אלה. השאלות בפרק מנוסחות אמנם במתכונת הדומה לזו של בחינות הבגרות אך רעיון התשובה שלנו בכל שאלה מונחה על פי הסעיפים לעיל, בפרט סעיפים א-ג.

שיקולים בבחירת טיפוס נתונים

כאשר אנו בוחרים במבנה נתונים יש שנוכח את היתרונות והחסרונות של כל אחד מהמבנים:

גודל	הוספת איבר	הוצאת איבר	גישה ישירה לכל איבר	
חסום	אפשרי אך מסורבל	אפשרי אך מסורבל	ניתן	מערך
לא חסום	אפשרי	אפשרי	לא ניתן	רשימה
לא חסום	רק מראש המחסנית	רק מראש המחסנית	לא ניתן	מחסנית
לא חסום	רק בסוף התור	רק מראש התור	לא ניתן	תור
לא חסום	אפשרי	אפשרי	לא ניתן	עץ בינרי

קיימים שיקולים חשובים נוספים מהם נציין שניים:

- יעילות הפתרון: כאשר כמות המידע הנשמרת גדולה נעדיף לבחור מבנה נתונים אשר סדר גודל של הפעולות טוב יותר.
- יתירות מידע: רצוי שמידע כלשהו לא יופיע יותר מפעם אחת.

עמוד	תבניות בשימוש	טנ"מ לפתרון	שאלה
154	4.1 בניה של רשימה. 4.5 מקום ברשימה. מציאת מקסימום. (*)	מערך רשימה	9.1 משחק השביל
156	4.1 בניה של רשימה. 4.4 מקום ברשימה. 4.5 מחיקת איברים מרשימה.	רשימה תלת-רשימה	9.2 טרמפיאדה
158	4.1 בניה של רשימה. 4.2 שרשור רשימות. 4.4 מקום ברשימה. 4.5 מחיקת איברים מרשימה.	מערך דו-מימדי רשימה	9.3 מערכת שעות
160	6.1 הכנס נתונים לתור. 6.2 הוצא נתונים מתור. 4.1 בניה של רשימה. 4.5 מחיקת איברים מרשימה.	תור רשימה	9.4 לוח מודעות
162	4.3 נמצא ברשימה. 4.4 מקום ברשימה. האם זוג פריטים סמוכים בסדרה מקיימים יחס נתון. (*) מניה. (*)	מערך חד-מימדי רשימה מערך דו-מימדי	9.5 ניווט חסמבה
166	6.1 הכנס נתונים לתור. 6.2 הוצא נתונים מתור.	תור	9.6 תור צומת
168	4.4 מקום ברשימה. בנית מספר מספרות בודדות. (*)	עץ בינארי מערך	9.7 קוד מורס
170	8.2 סריקה בסדר תחילי. 8.4 סריקה בסדר סופי. 6.1 הכנס נתונים לתור. 6.2 הוצא נתונים מתור. 5.5 איסוף בקיזוז.	עץ בינארי תור מחסנית	9.8 עץ ביטוי

(*) תבניות המופיעות בחוברת "תבניות במדעי המחשב".

שאלה 9.1 - משחק השביל (מבחן לדוגמה תשנ"ח)

במשחק-השביל מורכב לוח המשחק משביל של משבצות לאורכו מתקדמים השחקנים על ידי הטלת קוביה וצעידה של מספר משבצות בהתאם. מספר המשבצות הוא n , $30 \leq N \leq 500$. במשבצות מסוימות קיימים "מעקפים" המובילים למשבצות אחרות לאורך השביל (בכיוון ההתקדמות או בכיוון ההפוך). מספר המעקפים אינו ידוע. אם שחקן חונה במשבצת בה מתחיל מעקף עליו לעבור למשבצת אליה מוביל המעקף. בתחילת המשחק נמצאים שני השחקנים לפני המשבצת הראשונה. השחקן הפותח במשחק נבחר באופן אקראי. מנצח במשחק הוא השחקן הראשון שעובר את נקודת הסיום (לדוגמא: אם בתור מסוים נמצא שחקן 3 צעדים מנקודת הסיום, והקוביה הראתה את המספר 5, יסיים השחקן את המסלול בתור זה).

- א. הצע ייצוג מתאים למשחק-השביל, שיאפשר הגדרה של שביל כלשהו. בייצוג זה תהיה סיבוכיות המעבר במעקף $O(1)$. תאר את הייצוג במילים וכתוב הצהרות מתאימות בסביבת העבודה.
- ב. כתוב ממשק עברי למשחק-השביל. על הממשק לטפל בהגדרות לוח המשחק, השחקנים ומהלכי המשחק.
- ג. אורך מעקף הוא מספר המשבצות עליהן מדלג השחקן כאשר הוא עובר בין משבצת המוצא של מעקף למשבצת היעד שלו. כתוב אלגוריתם מילולי בשם **אורך-מעקף-מקסימלי** (שביל) המקבל שביל בהתאם לייצוג שבחרת בסעיף א ומחזיר את אורך המעקף המקסימלי הקיים בו.
- ד. מהי סיבוכיות האלגוריתם שכתבת בסעיף ב? נמק.

רעיון הפתרון

מאפייני הבעיה:

שחקן נדרש לנוע על משבצות השביל, לכן מבנה הנתונים המתאים למשחק הוא מבנה המאפשר גישה ישירה לכל משבצת במסלול.

נבחן ייצוגים שונים לטני"מ משחק-השביל:

ייצוג 1: השביל ייוצג על ידי מערך בגודל N שכל אחד מאיבריו הוא מטיפוס משבצת. טיפוס משבצת הוא מספר שלם. משבצת שאינה מעקף ערכה 0. משבצת שהיא מעקף ערכה חיובי או שלילי בהתאם למספר המשבצות שיש לנוע בכיוון מעלה השביל או מורדו.

ייצוג 2: השביל ייוצג על ידי רשימה שכל איבר בה הוא מטיפוס משבצת. לכל משבצת יש מצביע נוסף, המצביע למשבצת אליה מוביל המעקף.

הייצוג המתאים למשחק-השביל הינו רשומה/מבנה אשר שדותיו הם: **השביל** באחד הייצוגים הנבחרים, ושני משתנים אשר יכילו את מיקומי השחקנים ברגע נתון.

הממשק של משחק-השביל צריך להכיל פעולות המטפלות באתחול הלוח והשחקנים, זיהוי מקומו של כל שחקן, קידום שחקן למשבצת חדשה, וטיפול במעקפים.

נבחן את פעולות הממשק בכל אחד מהייצוגים האפשריים:

אתחל-לוח

בייצוג 1: נסרוק את המערך ונתאים לכל משבצת ערך מספרי מתאים. סיבוכיות הפעולה $O(n)$.

בייצוג 2: בתהליך הבניה של לוח המשחק ננהל מונה הקובע אם גודל המעקף אותו יש להתאים למשבצת אינו חורג מגבולות הלוח, במעקף יורד, והוספת מספר משבצות מתאים במעקף עולה. סיבוכיות הפעולה $O(n)$.

התבניות בהן נשתמש: תבנית 4.1 **בניה של רשימה.**

קידום שחקן (משחק-השביל, שחקן, צעדים)

בייצוג 1: חישוב גודל הקפיצה וגישה מיידית למשבצת המתאימה כולל המעקף אם קיים. סיבוכיות הפעולה $O(1)$.

בייצוג 2: קידום השחקן מהמיקום הנוכחי מספר צעדים רצוי. אם קיים מעקף במשבצת אליה הגיע, קפיצה למשבצת החדשה. סיבוכיות הפעולה $O(n)$.

התבניות בהן נשתמש: וריאציה של תבנית 4.4 **מקום ברשימה.**

אורך-מעקף-מקסימלי (משחק-השביל)

בייצוג 1: יש לסרוק את המערך ולהחזיר את ערך המשבצת בעלת הערך הגדול ביותר בערכה המוחלט. סיבוכיות הפעולה $O(n)$.

בייצוג 2: יש לסרוק את השביל ובכל פעם שמגיעים למעקף, יש לספור את גודלו ולהחזיר את הגודל המקסימלי. סיבוכיות הפעולה היא $O(n^2)$ כי עבור מעקף שלילי, במורד הלוח, יש לסרוק את לוח המשחק מתחילתו.

התבניות בהן נשתמש: שילוב של תבנית 4.4 **מקום ברשימה** עם תבנית **מציאת מקסימום** שבחוברת תבניות במדעי המחשב.

- מתוך השוואת כמות העבודה שיש לבצע בשני הייצוגים השונים, נראה שלייצוג על ידי מערך יש עדיפות ברורה על פני ייצוג ברשימה. אולם אם אין חסם ידוע מראש על גודלו של n (מספר המשבצות), אזי נבחר בייצוג של רשימה, כיון שייצוג באמצעות מערך מחייב חסם עליון לא גדול מדי על n .

■

- ביציאה מהעיר "שימון" יש טרמפיאדה. אנשים המעוניינים בטרמפ מגיעים לטרמפיאדה ומחכים. כאשר מכונית עוצרת בטרמפיאדה, מכריז הנהג על יעדו ועל מספר המקומות הפנויים במכוניתו. למכונית נכנסים אנשים לפי סדר העדיפויות הבא: קודם – חיילות, לאחר מכן – חיילים, ורק בסוף אזרחים. בתוך כל קבוצה (חיילות, חיילים, אזרחים) סדר הכניסה למכונית הוא לפי סדר ההגעה לטרמפיאדה. רוצים לבנות מודול תוכנה לניהול הטרמפיאדה.
- א. ייצג את המידע על האנשים שמחכים בטרמפיאדה.
- ב. כתוב אלגוריתם **אחזר-רשימת-נוסעים-ליעד** (טרמפיאדה, יעד, מספר מקומות פנויים). האלגוריתם מקבל יעד נסיעה ומספר מקומות פנויים במכונית, מחזיר רשימה של הנוסעים שייכנסו למכונית ומעדכן את המידע השמור על האנשים שמחכים בטרמפיאדה, בהתאם לייצוג שכתבת בסעיף הקודם.
- ג. למערכת התוכנה של הטרמפיאדה הוכנס מודול המטפל במסלולי הנסיעה מהעיר "שימון" ליעדים שונים. במודול ממומשת הפעולה **אחזר-יעד-ביניים** (יעד):

הפעולה מחזירה רשימה של יישובים שבין העיר "שימון" ובין היעד (כולל). הרשימה מסודרת לפי סדר היישובים לאורך מסלול הנסיעה.	אחזר-יעד-ביניים (יעד)
---	------------------------------

כתוב אלגוריתם **אחזר-רשימת-נוסעים** (טרמפיאדה, יעד, מספר מקומות פנויים במכונית). האלגוריתם מקבל יעד נסיעה ומספר מקומות פנויים במכונית ומחזיר רשימה של הנוסעים שייכנסו למכונית לפי הכלל הזה: תחילה ייכנסו למכונית אנשים המעוניינים להגיע ליעד של הנהג, לאחר מכן אנשים שיעדם הוא היישוב שלפני היעד של הנהג, וכך הלאה עד למילוי המקומות הפנויים במכונית. אם כמה אנשים מעוניינים להגיע לאותו יעד, תיקבע כניסתם למכונית לפי סוג הנוסע (חיילת, חייל, אזרח), ולפי סדר הגעתם לטרמפיאדה.

רעיון הפתרון

מאפייני הבעיה:

לצורך ניתוח הבעיה, נניח כי בטרמפיאדה יש N טרמפיסטים. לכאורה נראה שהייצוג המתאים לבעיה הוא מערכת של 3 **תורים**. בחירה במבנה כזה מעוררת בעיה כאשר יעד הנסיעה אינו מתאים לטרמפיסט הראשון בתור אך מתאים לאחד הטרמפיסטים שאחריו. ולכן נבחר במבנה נתונים רשימה, המאפשר פעולות של הוצאה והכנסה של איברים בכל מקום, בצורה קלה ונוחה.

נבחן ייצוגים שונים לטני"מ **טרמפיאדה**:

ייצוג 1: מכיוון שיש סדר קדימויות בין החיילות, החיילים והאזרחים, ובינם לבין עצמם נשמר סדר העלייה לטרמפ לפי סדר ההגעה, נוכל לייצג את הטרמפיאדה באמצעות **רשימה** שבה כל טרמפיסט מצטרף לסוף הרשימה.

ייצוג 2: נייצג את הטרמפיאדה **בתלת-רשימה**, כלומר, מערכת של 3 רשימות - אחת לחיילות, שנייה לחיילים ושלישית לאזרחים. הוספת טרמפיסט חדש תתבצע לסופה של הרשימה המתאימה.

נבחן את הפעולות הנדרשות לניהול הטרמפיאדה בכל אחד מהייצוגים השונים:

הכנסה של טרמפיסט לטרמפיאדה

ייצוג 1: הכנסה של טרמפיסט לרשימה תיעשה בסופה. סיבוכיות הפעולה $O(n)$.
התבנית בה נשתמש: תבנית 4.4 מקום ברשימה.

ייצוג 2: הכנסה של טרמפיסט תיעשה לסופה של הרשימה המתאימה. סיבוכיות הפעולה היא $O(n)$ אבל n הוא רק שליש ממספר הטרמפיסטים הכללי בטרמפיאדה.
התבנית בה נשתמש: תבנית 4.4 מקום ברשימה.

- ניתן לשפר את סיבוכיות הפעולה על ידי החזקת מצביע נוסף לאיבר האחרון ברשימה.

אחזר-רשימת-נוסעים-ליעד (טרמפיאדה, יעד, מספר מקומות פנויים)

ייצוג 1: הוצאה של טרמפיסט מהרשימה תעשה במקרה הגרוע ב-3 סריקות – סריקה ראשונה עבור החיילות, סריקה שנייה עבור החיילים וסריקה שלישית עבור האזרחים, שה"כ $3n$ צעדים. סיבוכיות הפעולה $O(n)$.

התבניות בהן נשתמש: תבנית 4.5 מחיקת איברים מרשימה, בשינוי קל לפיו המחיקה תתבצע בתנאי שעדיין יש מקום פנוי במכונית, ותבנית 4.1 בניה של רשימה לבניית הרשימה המוחזרת.

ייצוג 2: להוצאת טרמפיסטים, נסרוק תחילה את רשימת החיילות, ואחר כך, אם עדיין יש מקום פנוי במכונית, נסרוק את רשימת החיילים ולבסוף את רשימת האזרחים. במקרה הטוב נבצע עבודת סריקה על $n/3$ איברים ובמקרה הגרוע תתבצע העבודה על כל 3 הרשימות ונסרוק n טרמפיסטים. סיבוכיות הפעולה $O(n)$.

התבניות בהן נשתמש: תבנית 4.5 מחיקת איברים מרשימה, בשינוי קל לפיו המחיקה תתבצע בתנאי שעדיין יש מקום פנוי במכונית, ותבנית 4.1 בניה של רשימה לבניית הרשימה המוחזרת.

אחזר-רשימת-נוסעים (טרמפיאדה, יעד, מספר מקומות פנויים במכונית)

בהנחה שהפעולה אחזר-יעד-ביניים (יעד) החזירה רשימה שבה k יעדים, יש לבצע את הפעולה אחזר-רשימת-נוסעים-ליעד במקרה הגרוע k פעמים. סיבוכיות הפעולה $O(n*k)$.

- אמנם בשני הייצוגים הסיבוכיות דומה, אולם יש לזכור כי בפעולת ההוצאה של טרמפיסט מהטרמפיאדה, יש לבצע $3n$ פעולות בייצוג 1 ו- n פעולות בלבד בייצוג 2.

■

במכללת "סוד ההצלחה" הוחלט למחשב את מערכת השעות. לצורך כך הוגדר מודול מערכת שעות האוגר בתוכו את הנתונים הבאים: בכל יום מימי הלימודים א-ה ובכל שעה משעות הלימוד 08:00 – 16:00 נלמד קורס אחד. לכל שעה במערכת נאגרים הנתונים הבאים: מספר הקורס, מספר המורה המעביר את הקורס, ורשימה של מספרי הזיהוי של התלמידים הלומדים בקורס (מספר התלמידים בקורס אינו מוגבל). להלן חלק מממשק המודול **מערכת-שעות**:

<p>אחזר-רשימת-משתתפים (מערכת-שעות, מספר-קורס)</p>	<p>הפעולה מחזירה עותק של רשימת התלמידים בקורס שמספרו מספר-קורס. <u>הנחה</u>: מספר הקורס תקין.</p>
<p>אחזר-מורה (מערכת-שעות, מספר-קורס)</p>	<p>הפעולה מחזירה את מספר המורה בקורס שמספרו מספר-קורס. <u>הנחה</u>: מספר הקורס תקין.</p>
<p>אחזר-מספר-קורס (מערכת-שעות, יום, שעה)</p>	<p>הפעולה מחזירה את מספר הקורס המתקיים ביום יום ובשעה שעה. אם לא מתקיים קורס בזמן הנתון, יוחזר המספר 0. <u>הנחות</u>: היום והשעה תקינים.</p>

- יציג את המידע במודול **מערכת-שעות**.
- כתוב אלגוריתם המממש את הפעולה **אחזר-רשימת-משתתפים** (מספר-קורס).
- כאשר מורה חולה יש להודיע לכל המשתתפים בקורסים שבהם הוא מלמד באותו יום על ביטול השיעורים ליום זה. בנה פעולה **תלמידים-של-מורה** (מערכת-שעות, מספר-קורס, יום) המחזירה רשימה של מספרי הזיהוי של התלמידים שמלמד המורה שמספרו מספר-מורה ביום יום. יש לוודא שכל מספר זיהוי של תלמיד לא יופיע יותר מפעם אחת.
- מהי סיבוכיות הפעולה **תלמידים-של-מורה** כפונקציה של מספר התלמידים במכללה? נמק!

רעיון הפתרון

מאפייני הבעיה:

מנתוני הבעיה עולה כי במכללת "סוד ההצלחה" יש k קורסים, $1 \leq k \leq 40$. בכל קורס לומדים m תלמידים, $1 \leq m \leq n$, כאשר n מייצג את מספר התלמידים הכללי במכללה.

נבחן ייצוגים שונים לטנ"מ **מערכת-שעות**:

ייצוג 1: מערכת-השעות ייוצג על ידי מערך דו ממדי (מטריצה) שבו 5 שורות - שורה לכל יום, ו-8 עמודות - עמודה לכל שעה. כל איבר במטריצה יהיה מטיפוס שיעור, הכולל: מספר קורס, שם מורה ורשימת התלמידים המשוייכים לקורס.

ייצוג 2: מערכת-השעות תיוצג על ידי רשימה של קורסים. כל איבר ברשימה יכיל את המידע הבא: מספר קורס, שם מורה, יום, שעה, רשימת התלמידים המשוייכים לקורס.

ייצוג 3: מערכת-השעות תיוצג על ידי רשימה של תלמידים. כל איבר ברשימה יכול את פרטי התלמיד, ואת רשימת הקורסים אליהם משובץ התלמיד. כל קורס בייצוג זה יכול את: מספר הקורס, שם המורה, יום ושעת לימוד.

נבחן את פעולות הממשק בכל אחד משלושת הייצוגים האפשריים:

אחזר-רשימת-משתתפים (מערכת-שעות, מספר-קורס)

בייצוג 1: יש לסרוק את כל המטריצה כדי לאתר את הקורס, פעולה האורכת לכל היותר k צעדים, ולאחר מכן להחזיר את רשימת m התלמידים. אם מוחזרת הרשימה עצמה, סיבוכיות הפעולה $O(k)$ ואם מוחזר עותק של הרשימה, תהיה הסיבוכיות $O(k+m)$.

התבניות בהן נשתמש: תבניות 4.4 **מקום ברשימה** ו- 4.1 **בניה של רשימה**.

בייצוג 2: יש לסרוק את רשימת הקורסים כדי לאתר את הקורס המתאים, פעולה שאורכת לכל היותר k צעדים, ולאחר מכן לאחזר את רשימת התלמידים, כמו בייצוג הקודם. פעולה שסיבוכיותה היא $O(k+m)$.

התבניות בהן נשתמש: תבניות 4.4 **מקום ברשימה** ו- 4.1 **בניה של רשימה**.

בייצוג 3: יש לסרוק את רשימת n התלמידים, ולכל תלמיד יש לסרוק את רשימת הקורסים אליהם הוא משוייך, כל תלמיד משוייך לכל היותר ל- k קורסים. פעולה שסיבוכיותה $O(n*k)$.

התבניות בהן נשתמש: תבניות 4.4 **מקום ברשימה** ו- 4.1 **בניה של רשימה**.

- נראה שבשני הייצוגים הראשונים מתבצעת עבודה דומה בסיבוכיות $O(k)$ או $O(k+m)$, ואילו בייצוג השלישי מתבצעת עבודה בסיבוכיות $O(n*k)$. מכאן שיש עדיפות ברורה לשני הייצוגים הראשונים.

אחזר-מורה (מערכת-שעות, מספר-קורס)

בייצוג 1: יש לסרוק את כל המטריצה כדי לאתר את הקורס, פעולה האורכת לכל היותר k צעדים. סיבוכיות הפעולה $O(k)$.

התבנית בה נשתמש: תבנית 4.4 **מקום ברשימה**.

בייצוג 2: יש לסרוק את רשימת הקורסים כדי לאתר את הקורס המתאים, פעולה שאורכת לכל היותר k צעדים. סיבוכיות הפעולה $O(k)$.

התבנית בה נשתמש: תבנית 4.4 **מקום ברשימה**.

בייצוג 3: יש לסרוק את רשימת n התלמידים, לאתר תלמיד הלומד בקורס ולאחזר את שם המורה. במקרה הגרוע תהיה סיבוכיות הפעולה $O(n*k)$.

התבנית בה נשתמש: תבנית 4.4 **מקום ברשימה**.

- בשני הייצוגים הראשונים מתבצעת עבודה דומה בסיבוכיות $O(k)$, ואילו בייצוג השלישי מתבצעת עבודה בסיבוכיות $O(n*k)$. מכאן שיש עדיפות ברורה לשני הייצוגים הראשונים.

אחזר-מספר-קורס (מערכת-שעות, יום, שעה)

בייצוג 1: זוהי פעולה הכרוכה בגישה ישירה לשורה ולעמודה המתאימים ואיחזור המידע. פעולה שסיבוכיותה $O(1)$.

בייצוג 2: יש לעבור על רשימת הקורסים, לאתר קורס המתאים ליום ולשעה הנתונים, ולאחזר את מספרו. פעולה החיפוש משתמשת בתבנית איתור ברשימה, שסיבוכיותה $O(k)$.
התבנית בה נשתמש: תבנית 4.4 מקום ברשימה.

בייצוג 3: הפעולה אינה פעולה טבעית לייצוג זה. יש לעבור על רשימת התלמידים, לאתר אצל אחד התלמידים קורס המתאים ליום ולשעה הנתונים ולהחזיר את מספרו. פעולה שסיבוכיותה, במקרה הגרוע $O(n*k)$.

התבנית בה נשתמש: תבנית 4.4 מקום ברשימה תחילה לרשימת התלמידים ואחר כך למציאת הקורס המתאים ברשימת התלמיד.

- קיימת עדיפות ברורה לייצוג 1, שסיבוכיותו הנמוכה ביותר.

תלמידים-של-מורה (מערכת-שעות, מספר-קורס, יום)

בייצוג 1: יש לגשת ישירות לשורה המתאימה במטריצה, לסרוק את הקורסים לאותו יום, לכל היותר 8 קורסים, ולשרשר ללא כפילויות את רשימות התלמידים המתאימים לאותו מורה. פעולת השרשור אורכת $8*m$ צעדים, כלומר סיבוכיות $O(m)$. (סיבוכיות ההכנסה לרשימה בשני הייצוגים נדונה בפרק סיבוכיות של פעולות). לאחר מכן יש לבטל כפילויות, פעולה בסיבוכיות ריבועית.

אם למורה מסויים משוייכים כלל תלמידי המכללה, תהיה הפעולה בסיבוכיות $O(n^2)$.
התבניות בהן נשתמש: תבניות 4.2 שרשור רשימות ו-4.5 מחיקת איברים מרשימה.

בייצוג 2: יש לסרוק את רשימת הקורסים ולאחר את כל הקורסים המועברים על ידי המורה. עבור כל קורס כזה, יש לאחזר את רשימת התלמידים ולשרשר אותה לרשימת התלמידים של המורה. בדומה לייצוג הקודם, פעולת הסריקה ופעולת ביטול הכפילויות נעשים בסיבוכיות $O(n^2)$.
התבניות בהן נשתמש: תבניות 4.4 מקום ברשימה, 4.2 שרשור רשימות ו-4.5 מחיקת איברים מרשימה.

בייצוג 3: יש לסרוק את רשימת התלמידים. עבור כל תלמיד יש לסרוק את רשימת הקורסים שלו ולבדוק האם הוא תלמיד של המורה. הרשימה נבנית ללא כפילויות, ולכן סיבוכיותה $O(n*k)$. מכיוון שמספר הקורסים k מוגבל ל-40, ניתן לטעון שסיבוכיות הפעולה ליניארית.
התבניות בהן נשתמש: 4.4 מקום ברשימה, 4.1 בניה של רשימה.

- למרות שלכאורה הייצוג השלישי נחשב לייצוג גרוע עבור פעולות הממשק, הרי שבפעולה תלמידים-של-מורה יש לו עדיפות ברורה על פני שני הייצוגים הראשונים. השיקול הסופי בבחירת מבנה הנתונים תלוי בפעולות המתבצעות בשכיחות הגבוהה ביותר. אם במהלך העבודה מרבים באחזור מידע ושימוש בפעולות מסוג פעולות הממשק, נעדיף את הייצוג הראשון או השני, ואם מרבים בפעולות מהסוג של טיפול בתלמידים לפי המורה המלמד אותם, נעדיף את הייצוג השלישי.

■

שאלה 9.4 - לוח מודעות

טיפוס הנתונים **לוח-מודעות** משמש כמקום אשר עליו תולים הודעות המגיעות בסדר כרונולוגי. מספר ההודעות בלתי ידוע מראש ובלתי מוגבל. ידוע גם **שלוח-המודעות** מכיל מקום אליו ניתן להעביר ולתלות את כל ההודעות שתוקפן פג. כל הודעה מאופיינת ע"י תאריך ההודעה וטקסט ההודעה. הודעה נחשבת כהודעה שתוקפה פג אם עבר מספר מסוים של ימים מאז נתלתה ההודעה על הלוח. להלן חלק מהפעולות המוגדרות על טיפוס הנתונים **לוח-מודעות**:

פעולה המקבלת שני תאריכים תאריך-1 ו- תאריך-2 ומחזירה את מספר הימים שעבר בין שני תאריכים אלה. הנחה : התאריכים תקינים.	מספר-הימים (תאריך-1 , תאריך-2)
פעולה המקבלת הודעה ומוסיפה אותה להודעות התלויות על לוח-המודעות . הנחה : הלוח מאותחל.	תלה-הודעה (לוח-מודעות , הודעה)
פעולה המעבירה את כל ההודעות שתוקפן פג אל האזור המיועד לכך ב לוח-המודעות . מודעה שתוקפה פג היא מודעה שתאריך התליה שלה גדול ב- N ימים מהתאריך של היום. הנחה : הלוח מאותחל.	העבר-הודעות-ישנות (לוח-מודעות , היום , N)

- א. ייצג את טיפוס הנתונים לוח-מודעות.
ב. כתוב אלגוריתם המממש את הפעולה: **תלה-הודעה** (**לוח-מודעות**, **הודעה**).

רעיון הפתרון

מאפייני הבעיה:

הפעולות החשובות על לוח המודעות הן פעולות של הוספה של מודעות והסרה של מודעות ישנות (העברה ללוח השני). ניתן להתייחס אל לוח המודעות כאל לוח "שקוף" בו ניתן לקרוא את כל המודעות אולם לא ניתן להוסיף או להסיר מודעה מהאמצע. מכיוון שהמודעות מופיעות בסדר כרונולוגי, מתוספות מודעות חדשות לסוף הלוח, ובתחילת הלוח מצטברות המודעות הישנות.

נבחן ייצוגים שונים לטנ"מ **לוח-מודעות**:

ייצוג 1: לוח המודעות מיוצג על ידי מערכת של שני תורים, שכל אחד מאיבריו הוא מטיפוס מודעה. תור אחד הוא תור המודעות הפעילות, תור שני הוא תור המודעות שפג תוקפן.

ייצוג 2: לוח המודעות מיוצג על ידי מערכת של שתי רשימות, שכל אחד מאיבריה הוא מטיפוס מודעה. רשימה אחת היא רשימת המודעות הפעילות. רשימה שנייה היא רשימת המודעות שפג תוקפן.

טיפוס המודעה הינו רשומה/מבנה המכילים שדה תאריך ושדה טקסט.

נבחן את פעולות הממשק בכל אחד משני הייצוגים האפשריים :

תלה-הודעה (לוח-מודעות, הודעה)

ייצוג 1: מודעה חדשה מתווספת לסופו של התור. סיבוכיות הפעולה $O(1)$.

ייצוג 2: מודעה חדשה מתווספת לסופה של הרשימה. סיבוכיות הפעולה $O(n)$.

העבר-הודעות-ישנות (לוח-מודעות, היום, \mathbb{N})

בייצוג 1: יש לרוקן את התור מכל המודעות שהתאריך שלהן קודם לתאריך הנתון ולהעבירן לסופו של התור של המודעות שתוקפן פג. סיבוכיות הפעולה $O(n)$.
התבניות בהן נשתמש: תבנית 6.2 הוצא נתונים מתור ו- 6.1 הכנס נתונים לתור.

בייצוג 2: יש לרוקן את הרשימה מכל המודעות שהתאריך שלהן קודם לתאריך הנתון ולהעבירן לסופה של הרשימה של מהודעות שפג תוקפן. סיבוכיות הפעולה במימוש רשימה בשרשרת חוליות היא $O(n)$ (יעילות ההסרה של איברים מתחילת רשימה נעשה בפרק 7 סיבוכיות של פעולות). סיבוכיות הפעולה במימוש רשימה במערך היא $O(n^2)$.
התבניות בהן נשתמש: שילוב של תבנית 4.5 מחיקת איברים מרשימה עם תבנית 4.1 בניה של רשימה.

מספר-הימים (תאריך-1, תאריך-2)

זוהי פעולה חישובית שאינה תלויה בייצוג של לוח המודעות. אלגוריתם לחישוב הפרש בין שני תאריכים נמצא בפרק 3 יחידת ספריה ופרויקט ב-C.

- מתוך השוואת כמות העבודה שיש לבצע בשני הייצוגים השונים, נראה שלייצוג על ידי תור יש עדיפות ברורה על פני ייצוג ברשימה.

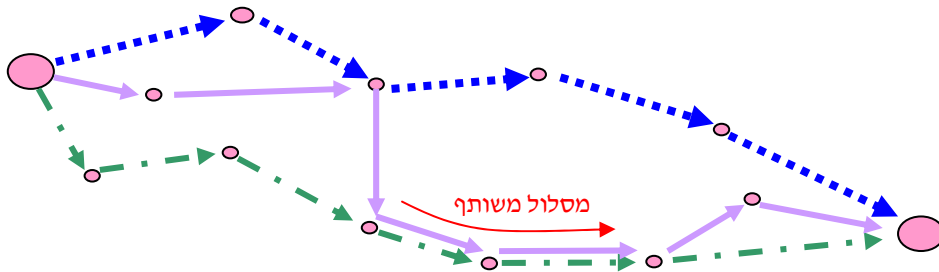
■

שאלה 9.5 - ניווט חסמבה (מקור: עליזה נחמני)

בתרגיל ניווט שבוצע במחלקה הסודית "חסמבה", חולקה המחלקה ל-7 חוליות. כל האתרים בתרגיל מוספרו מ-1 עד N . $10 \leq N \leq 50$. כל חוליה קיבלה רשימה של אתרים שבהם היא צריכה לבקר (לפי סדר הופעתם ברשימה). כל מסלולי הניווט מתחילים באתר שמספרו 1 ומסתיימים באתר "המערה החשמלית" שמספרה N . חוליה אינה מבקרת באתר כלשהו יותר מפעם אחת.

דוגמא למסלולים: מסלולה של חוליה j: $\rightarrow 1, 13, 7, 9, 5, 14, 25$
 מסלולה של חוליה k: $\rightarrow 1, 2, 7, 14, 9, 5, 18, 25$
 לחוליות j ו-k "מסלול משותף" העובר באתרים: 9, 5

"מסלול משותף" הוא רצף (גדול מ-1) של אתרים הנמצאים במסלול של שתי חוליות. נקודת מפגש בודדת הנמצאת במסלול של שתי חוליות אינה נחשבת "קטע משותף".



תמר שהיא בוגרת עיצוב תכנה, נעזרה במחשב להכנת תרגיל הניווט. להלן חלק מהממשק העברי לטיפוס הנתונים המופשט **ניווט**:

אתחל-ניווט	פעולה המתחלת את טיפוס הנתונים ניווט.
אתר-במסלול? (ניווט, חוליה, אתר)	פעולה המקבלת חוליה ושם של אתר, ומחזירה "אמת" אם האתר נמצא במסלול הניווט של החוליה, ו"שקר" אחרת. <u>הנחה</u> : טיפוס הנתונים ניווט מאותחל.
מסלול-משותף (ניווט, חוליה-1, חוליה-2)	פעולה המקבלת מספרי שתי חוליות ומחזירה רשימה של שמות כל האתרים הנמצאים במסלול המשותף לשתייהן. הערה: מסלול משותף הוא רצף של אתרים הנמצאים בשני מסלולים. <u>הנחה</u> : טיפוס הנתונים ניווט מאותחל.

- הצע ייצוג מתאים לטיפוס הנתונים המופשט **ניווט**.
- המפקד ירון זהבי מעוניין לדעת כמה מהקבוצות יבקרו באתר מסוים. כתוב אלגוריתם המקבל שם של אתר ומחזיר רשימה של מספרי החוליות העתידות לבקר בו.
- כתוב אלגוריתם בסביבת העבודה המממש את הפעולה: **מסלול-משותף** (ניווט, חוליה-1, חוליה-2).

רעיון הפתרון

מאפייני הבעיה:

מספר האתרים בהם תבקר כל חוליה אינו ידוע. לכאורה יכולה חוליה לבקר בכל N האתרים. מספר החוליות הינו בדיוק 7.

נבחן ייצוגים שונים לטני"מ ניווט:

ייצוג-1: קיימות 7 חוליות ולכן נוכל לייצג את הניווט על ידי מערך בגודל 7 שכל איבר בו הוא רשימת האתרים שבו תבקר החוליה. אינדקס המערך ייצג את מספרה של החוליה. ניתן לייצג את החוליות ברשימה של רשימות האתרים, אולם במקרה זה נצטרך להחזיק מודול מיוחד לכל סוג רשימה, האחד לטיפוס הנתונים חוליה והשני לטיפוס הנתונים אתר.

ייצוג-2: מספר האתרים חסום על ידי N , לכן נייצג את מערכת הניווט על ידי מערך דו-מימדי בגודל $7*N$, שבו כל שורה מייצגת חוליה וכל עמודה מייצגת אתר. המידע על האתר יכיל ערך בוליאני ומספר. אם חוליה i מבקרת באתר j , ערך התא המתאים יהיה 'אמת' וערך המספר יהיה מספר טבעי המציין את מקומו של האתר ברשימת הביקורים של החוליה. ואם אינה מבקרת באתר יהיה ערך התא 'שקר'.

נבחן את פעולות הממשק בכל אחד משני הייצוגים האפשריים:

אתר-במסלול? (ניווט, חוליה, אתר)

ייצוג 1: יש לסרוק את רשימת החוליה. סיבוכיות הפעולה $O(n)$.
התבנית בה נשתמש: תבנית 4.3 **נמצא ברשימה.**

ייצוג 2: גישה ישירה לתא במערך. סיבוכיות הפעולה $O(1)$.

מסלול-משותף (ניווט, חוליה-1, חוליה-2)

ייצוג 1: עבור כל אתר בחוליה-1 נבדוק אם **האתר-במסלול?** של חוליה-2. כשימצא אתר נבדוק האם קיימת התקדמות במקביל בשתי החוליות. סיבוכיות הפעולה היא $O(n^2)$.
התבניות בהן נשתמש: עבור כל איבר ברשימה מתבצעת התבנית 4.3 **נמצא ברשימה?** אם הערך המוחזר הוא 'אמת', נשתמש בתבנית 4.4 **מקום ברשימה** לבדיקת המשך המסלול. ניתן לוותר על הפעולה **אתר-במסלול?** ולבצע רק את התבנית 4.4 **מקום ברשימה** דבר שיחסוך סריקה כפולה של הרשימה.

ייצוג 2: עבור כל אתר בחוליה-1, ניגש ישירות לתא במערך של חוליה-2 ונבדוק האם הוא במסלול. אם במסלול, נבדוק האם קיימת התקדמות במקביל תוך חזרה על אותה פעולה עבור האתר הבא. הפעולה מצריכה סריקת רשימה אחת בלבד וגישה ישירה לאיברים ברשימה השניה, ולכן סיבוכיות הפעולה היא $O(n)$.

התבניות בהן נשתמש: וריאציה של התבנית **האם זוג פריטים סמוכים בסדרה מקיים יחס נתון** משולב עם תבנית **מניה**. שתי התבניות מוצגות בחוברת "תבניות במדעי המחשב".

מספר-המבקרים-באתר (ניווט, אתר)

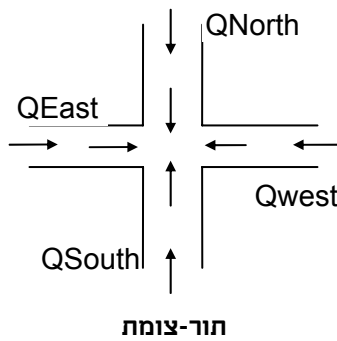
ייצוג 1: עבור כל אחת מ-7 החוליות נבצע את הפעולה **אתר-במסלול**? שסיבוכיותה נדונה קודם, ולכן סיבוכיות הפעולה $O(n)$.

ייצוג 2: יש לסרוק את העמודה המתאימה במערך - בדיוק 7 צעדים, ולכן סיבוכיות הפעולה $O(1)$.
התבנית בה נשתמש: תבנית מניה המוצגת בחוברת "תבניות במדעי המחשב".

- מהיבט של **יעילות זמן ריצה** - ייצוג 2 יעיל יותר מייצוג 1. מהיבט של **יעילות של מקום** - ייצוג 2 הינו ייצוג בזבזני המחייב שמירה של מטריצה שגודלה $7*N$. אם מספר האתרים שבו תבקר כל חוליה מתקרב ל- N נעדיף את ייצוג 2, אחרת נעדיף את ייצוג 1.

■

שאלה 9.6 - תור-צומת (מקור לא ידוע)



צומת כבישים מכיל 4 כיוונים כמתואר בשרטוט. רכב הנכנס בכיוון אחד של הצומת יצא ממנו רק לאחר שכל כלי הרכב שנכנסו לפניו באותו כיוון יפנו את הצומת.

ניתן להכניס רכב או להוציא רכב מכל אחד מארבעת הכיוונים.

להלן הממשק העברי לטיפוס הנתונים **תור-צומת**:

אתחל-תור-צומת	הפעולה מחזירה תור-צומת שארבעת כיווניו ריקים.
הכנס-לתור-צומת (Jq, direction, x)	הפעולה מכניסה את האיבר x, לתור-צומת Jq בכיוון direction. <u>הנחה</u> : Jq מאותחל.
הוצא-מתור-צומת (Jq, direction)	הפעולה מקבלת תור-צומת וכיוון התחלתי direction. הפעולה מחזירה את האיבר שהוצא מתור-צומת, בצורה הבאה: אם קיים איבר בתור שכיוונו direction, יוחזר איבר זה. אחרת יוחזר האיבר מהצד הבא בכיוון השעון, שאינו ריק. <u>הנחות</u> : Jq מאותחל ולא ריק.
תור-צומת-ריק? (Jq)	הפעולה מחזירה 'אמת' אם כל בכיוונים בתור-צומת ריקים, ו-'שקר' אחרת. <u>הנחה</u> : Jq מאותחל.

א. ייצג את טיפוס הנתונים **תור-צומת**.

ב. כתוב אלגוריתם למימוש הפעולה: **הוצא-מתור-צומת** (Jq, direction).

ג. מהי סיבוכיות האלגוריתם שכתבת בסעיף ב?

ד. כתוב אלגוריתם בשם **אזן-תור-צומת** (Jq, Newq) תוך שימוש בפעולות הממשק **תור-צומת** בלבד, המקבל תור-צומת ומעביר את כל הממתינים בין הכיוונים השונים, כך שבסיום הפעולה יכיל כל אחד מארבעת הכיוונים מספר "זזהה" של כלי רכב. אם מספר כלי הרכב הכולל אינו מתחלק ב-4 מותר שיהיה הפרש של 1 בין ארבעת הכיוונים.

רעיון הפתרון

מאפייני הבעיה:

הבעיה מציגה הדמיה של כלי רכב המגיעים ונכנסים לצומת בעל 4 כיוונים תחת אילוצים מסוימים. הגעת רכב בכיוון מסוים, כניסה ופינוי הצומת נעשים לפי כלל FIFO ולכן כל כיוון בצומת ייוצג על ידי תור, והצומת כולו ייוצג על ידי מערכת של ארבעה תורים בעלי סדר מעגלי.

נבחן ייצוגים שונים לטיפוס הנתונים **תור-צומת**:

ייצוג 1: מערך בגודל 4 שכל איבר בו הוא תור. אינדקס התא ייצג כיוון. נגדיר באופן שרירותי שצד מסוים יהיה צד 0, זה שלאחריו בכיוון השעון יהיה צד 1 ואחריהם צדדים 2 ו-3. צד 0 עוקב לצד 3.

ייצוג 2: רשומה המכילה 4 שדות שכל אחד מהם הוא תור. כל תור יקבל שם לפי רוחות השמים, כאשר סדר הצמתים יהיה מעגלי - צפון, מזרח, דרום, מערב.

נבחן את פעולות הממשק בכל אחד משני הייצוגים האפשריים:

אתחל-תור-צומת

בפעולת האתחול בשני הייצוגים נשתמש בפעולת הממשק לתור **אתחל-תור** שתבוצע עבור כל אחד מ-4 הכיוונים. סיבוכיות הפעולה אתחל-תור היא $O(1)$ ולכן הפעולה אתחל-תור-צומת תהיה אף היא בסיבוכיות $O(1)$.

הכנס-לתור-צומת (Jq, direction, x)

בשני הייצוגים נעזר בפעולת הממשק לתור **הכנס-לתור** שסיבוכיותה $O(1)$ ולכן הפעולה הכנס-לתור-צומת תהיה אף היא בסיבוכיות $O(1)$.

הוצא-מתור-צומת (Jq, direction)

בשני הייצוגים נעזר בפעולת הממשק לתור **הוצא-מתור**. סדר גודל של פעולת ההוצאה היא $O(1)$ בייצוג דינאמי של שרשרת חוליות ו- $O(n)$ בייצוג סטטי של מערך. בבעיה שלפנינו אם התור ריק יש לעבור לתור הבא בכיוון השעון. כיון שיש בדיוק 4 תורים, הרי שגם פעולת הוצא-מתור-צומת תהיה $O(1)$ או $O(n)$ בהתאם למימוש.

אזן-תור-צומת (Jq, Newq)

בפעולת האיזון אנו נדרשים לבצע העברות בין התורים כך שמספר האיברים בכל תור יהיה דומה עד כדי הפרש של 1. פעולת ההעברה מרוקנת חלקית תורים עמוסים ומכניסה איברים לתורים עמוסים פחות. בהנחה שיש להעביר N איברים בין התורים, תוך שימוש בפעולות הממשק האחרות, הרי שסיבוכיות הפעולה תלויה בסיבוכיות פעולות הממשק.

התבניות בהן נשתמש: תבנית 6.1 **הכנס נתונים לתור** ו-6.2 **הוצא נתונים מתור**.

- בשני הייצוגים התקבלה סיבוכיות זהה. כיון שמימוש הפעולות **הוצא-מתור-צומת** ו**אזן-תור-צומת** פשוטים יותר במימוש של מערך של תורים נעדיף להשתמש בייצוג זה.

■

שאלה 9.7 - קוד מורס

בעבר היו מעבירים תשדורת אלחוטית באמצעות קוד מורס. כל אות בקוד זה מיוצגת על ידי צירוף של קווים ונקודות. במהלך התשדורת מופיע תו רווח בודד בסיום כל אות ורווח כפול בסיומה של מילה. להלן טבלת הסימנים המתאימים לאותיות ה-ABC:

A	. _	B	_ ...	C	_ . _ .	D	_ ..	E	.	F	.. _ .
G	__ .	H	I	..	J	. _ _ _	K	_ . _	L	. _ . .
M	__	N	_ .	O	_ _ _	P	. _ . .	Q	_ _ . _	R	. _ .
S	...	T	_	U	.. _	V	.. _ _	W	. _ _	X	_ . . _
Y	_ . _ _	Z	_ _ . .								

- הצע ייצוג מתאים לטיפוס הנתונים קוד-מורס.
- כתוב פעולה המקבלת קוד-מורס ומחזירה את מספר האותיות שהקוד שלהן מורכב מ-4 סימנים בדיוק.
- כתוב פעולה המקבלת קוד-מורס ומדפיסה בשורה ראשונה את האותיות שאורכן 1, בשורה שניה את האותיות שאורכן 2, בשורה השלישית את האותיות שאורכן 3 ובשורה האחרונה את האותיות שאורכן 4.
- כתוב פעולה הקולטת סימנים בקוד מורס, מפענחת ומדפיסה את ההודעה שהתקבלה. הנח שאין תווי רווח בהודעה פרט לאלה המציינים סוף אות או סוף מילה.

רעיון הפתרון

מאפייני הבעיה:

קוד מורס מכיל שני סימנים: קו ונקודה החוזרים על עצמם בצירופים שונים. מספר הסימנים מוגבל בגודלו. מתיאור הבעיה נובע שיש פעולות אחזור בלבד.

נבחן ייצוגים שונים לטני"מ קוד-מורס:

ייצוג 1:

קוד מורס מיוצג על ידי עץ בינארי. נקבע באופן שרירותי שסימן קו יפנה אותנו לבן השמאלי בעץ וסימן נקודה לבן הימני. כל צומת בעץ יכיל את האות המתאימה למסלול המתקבל משורש העץ ועד לצומת. שורש העץ אינו מייצג אות.

ייצוג 2:

ידוע שלכל אות בקוד-מורס יש לכל היותר 4 סימנים. ניתן לייצג את קוד המורס באמצעות טבלה, המכילה את התווים באופן הבא: את קוד המורס של כל אות נמיר למספר בינארי. נקבע באופן שרירותי שנקודה תיוצג על ידי הספרה 0 וקו על ידי 1. בתחילת כל מספר בינארי נוסיף את הספרה 1 כדי לשמור על אפסים מובילים באותיות המתחילות בנקודה. את המספר המתקבל במסלול הפענוח נמיר למספר בבסיס עשר שיהווה את אינדקס התא במערך המכיל את האות המתאימה לפענוח.

לדוגמא: האות C שערכה הבינארי הוא 1010 תיוצג על ידי 11010 שערכו העשרוני 26. בתא 26 במערך תאוחסן האות C. האות F שערכה הבינארי הוא 0010 תיוצג על ידי 10010 ותאוחסן בתא 18 במערך. נשים לב שלא הוספת 1 בתחילת המספר היה מתקבל עבור האות F (0010) ועבור האות R (010) ערך מספרי זהה הזה לערכה של האות N (10).

נבחן את הפעולה הנדרשת בייצוגים השונים:

הדפסה-ממוינת-לפי-מספר-סימנים-לאות (קוד-מורס)

בייצוג 1: סריקה העץ לרוחב תשיג את המטרה הרצויה. סיבוכיות הפעולה $O(n)$. הוא מספר הצמתים בעץ. **התבנית בה נשתמש:** 8.5 סריקה לרוחב.

בייצוג 2: הדפסת המערך מתחילתו, תוך דילוג על תאים ריקים, תיתן את ההדפסה הרצויה. סיבוכיות הפעולה $O(n)$. מייצג את גודל המערך שגדול ממספר האותיות ב-ABC.

- בשני הייצוגים מתקבלת סיבוכיות זהה. נעדיף את ייצוג 2 משום פשטותו.

פענח-אות (קוד-מורס, מחרוזת)

בייצוג 1: יש להתקדם במסלול בעץ לפי תווי המחרוזת הנתונה ולאחזר את ערך הצומת אליו הגענו כשמתקבל תו רווח. סיבוכיות הפעולה היא $O(n)$. מייצג את מספר התווים במחרוזת. **התבנית בה נשתמש:** וריאציה של התבנית 4.4 **מקום ברשימה** הפועלת על עץ בינארי (התקדמות נעשית לאורך מסלול בעץ).

בייצוג 2: יש לסרוק את התווים במחרוזת הקלט וליצור מהם מספר בינארי. לאחר מכן יש להמיר את המספר לערך עשרוני ולגשת ישירות לתא המתאים במערך. סיבוכיות הפעולה היא $O(n)$. מייצג את מספר התווים במחרוזת. **התבנית בה נשתמש:** **בנית מספר מספרות בודדות** המוצגת בחוברת "תבניות במדעי המחשב".

- בשני הייצוגים מתקבלת סיבוכיות זהה. ייצוג על ידי מערך כרוך בהחזקת מערך שלא כל איבריו מכילים ערך ובחישוב מורכב. נעדיף את ייצוג 1, עץ בינארי, משום פשטותו.

- מהיבט של **יעילות זמן ריצה** – עבור פיענוח השדר יהיה ייצוג 1 פשוט יותר מייצוג 2. עבור הדפסת התווים לפי אורך הקוד, יש עדיפות לפשטות של ייצוג 2. מהיבט של **יעילות של מקום** - ייצוג 2 הינו ייצוג בזבזני המחייב שמירה של מערך שחלק מתאיו אינו מנוצל. כיון שמרבית העבודה המתבצעת בקוד מורס היא עבודה של קידוד ופיענוח, נעדיף את ייצוג 1, ייצוג העץ, המאפשר ביצוע העבודה בצורה פשוטה יותר.

■

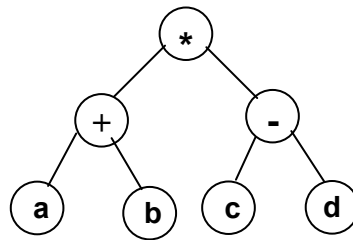
שאלה 9.8 - חישוב עץ ביטוי

הדרך המקובלת לייצוג ביטוי חשבוני היא בשיטת **infix**. בשיטת כתיבה זו נמצאת הפעולה החישובית בין שני הערכים עליהם היא פועלת. החיסרון של שיטת רישום זו הוא הצורך בסוגריים על מנת לשמור על סדר הקדימויות המתמטי של החישובים.

ניתן לייצג את הביטוי חשבוני בשיטת **postfix** לפיה נרשמת הפעולה אחרי הערכים עליהם היא פועלת, או בשיטת **prefix** לפיה נרשמת הפעולה לפני הערכים עליהם היא פועלת. שיטות אלו שומרות על סדר הקדימויות המתמטי ללא צורך בסוגריים.
לדוגמא:

ביטוי infix	ביטוי prefix	ביטוי postfix
$a + b$	$+ a b$	$a b +$
$(a + b) * (c - d)$	$* + a b - c d$	$a b + c d - *$

ניתן לשמור ביטוי חשבוני בעץ-ביטוי שבו כל צומת שאינו עלה מכיל פעולה חישובית, וכל עלה מכיל ערך. לדוגמא: הביטוי: $(a + b) * (c - d)$ ייוצג בעץ-ביטוי באופן הבא:



סריקה של עץ-ביטוי בסדר תחילי תיתן ביטוי prefix.

סריקה של עץ-ביטוי בסדר תוכי תיתן ביטוי infix.

סריקה של עץ-ביטוי בסדר סופי תיתן ביטוי postfix.

א. כתוב אלגוריתם מילולי המקבל כפרמטר עץ-ביטוי T ומחזיר את ערך הביטוי המאוחסן בצמתי העץ בחישוב postfix.

ב. כתוב אלגוריתם מילולי המקבל כפרמטר עץ-ביטוי T ומחזיר את ערך הביטוי המאוחסן בצמתי העץ בחישוב prefix.

רעיון הפתרון

מאפייני הבעיה:

ניתן לחשב את הביטוי באופן רקורסיבי ישירות מתוך העץ. דרך חישוב זו מוצגת בספר "עיצוב תכנה" בפרק עצים בינאריים.

נציג אלגוריתם לחישוב לא רקורסיבי, הנעזר במבני נתונים תור ורשימה. בשלב הראשון נסרוק את העץ לתוך תור, ובשלב השני נקרא את הנתונים מתוך התור ונחשב את הביטוי בעזרת מחסנית.

חישוב עץ-ביטוי בסדר סופי (T)

נסרוק את העץ בסדר סופי לתוך תור. בסריקה זו ישמרו הנתונים בתור כביטוי postfix. כדי לחשב את הביטוי, נוציא את האיברים בזה אחר זה מהתור. איבר שהוא ערך יוכנס למחסנית. איבר שהוא פעולה יגרום לשליפת שני האיברים שנמצאים בראש המחסנית, ביצוע החישוב עליהם ודחיפת התוצאה למחסנית. אם הביטוי תקין מובטח לנו שבסיום הפעולה יישאר במחסנית איבר אחד ויחיד וזוהי התוצאה. הפעולה עוברת על כל נתוני העץ פעמיים, פעם אחת בסריקה ופעם אחת בהוצאה מהתור. בתהליך החישוב, יש לבצע שליפה ודחיפה של איברים למחסנית כמספר פעולות החישוב שבעץ. מספר הפעולות המרבי בעץ ביטוי קטן ב-1 מ-2 בחזקת גובה העץ. גובהו של עץ בינארי בעל n צמתים יהיה בין $\log_2 n$ עבור עץ בינארי מלא (שכל רמה בו מלאה) ל- n עבור "עץ שרוך", לכן סורקים לכל היותר כל איבר 3 פעמים, ומכאן שסיבוכיות הפעולה היא $O(n)$.

התבניות בהן נשתמש: תבנית 8.4 סריקה בסדר סופי, 6.1 הכנס נתונים לתור, 6.2 הוצא נתונים מתור, 5.5 איסוף בקיזוז.

טיפוס המידע בתור: כל איבר בתור יהיה רשומה/מבנה המכילים פעולה או ערך.
טיפוס המידע במחסנית: ערכים מספריים.

חישוב עץ-ביטוי בסדר תחילי (T)

נסרוק את העץ בסדר תחילי לתוך תור. בסריקה זו ישמרו הנתונים בתור כביטוי prefix. כדי לחשב את הביטוי, נוציא את האיברים בזה אחר זה מהתור. אם האיבר שהוצא מהתור הוא פעולה נדחוף אותו למחסנית. ואם הוא מספר, נבדוק את האיבר שבראש המחסנית. אם זוהי פעולה, נדחוף את האיבר למחסנית. אם הוא מספר, אזי אם הביטוי החשבוני תקין, חייבת להיות מתחת למספר זה פעולה, לכן נשלוף את המספר ואת הפעולה, נבצע עליהם את החישוב, ונבדוק שוב את התוצאה מול האיבר שבראש המחסנית. בסיום התהליך, אם הביטוי החשבוני תקין, מובטח שתתקבל מחסנית ריקה והתוצאה החשובית האחרונה שחישבנו הינה תוצאת הביטוי.

סיבוכיות הפעולה היא $O(n)$ כי עוברים על כל איבר פעמיים בתהליך הסריקה מעץ לתור ומתור למחסנית, ובתהליך החישוב במחסנית עוברים על כל האיברים לכל היותר פעמיים, פעם אחת בהכנסה למחסנית ופעם אחת בשליפה ממנה. יתכן ויהיו איברים שלא יוכנסו למחסנית כלל ובמקומם תוכנס תוצאת חישוב שנעשתה עליהם.

התבניות בהן נשתמש: תבנית 8.2 סריקה בסדר תחילי, 6.1 הכנס נתונים לתור, 6.2 הוצא נתונים מתור, 5.5 איסוף בקיזוז.

טיפוס המידע בתור ובמחסנית: כל איבר יהיה רשומה/מבנה המכילים פעולה או ערך.

■

