

"אם זכרוני אינו מטעני, אני לא זוכר" (עמוס שוסט)

מערך שיעור בנושא מבני נתונים

יותם אלירז

מורן ניסן

הקדמה

בשיעור הראשון על מספרים בינאריים, הבנו שנתון בודד במחשב נשמר בצורת מספר בינארי, מספר המורכב מאפסים ומאחדים, כאשר 0 או 1 מיוצגים במחשב על ידי זרם חשמלי.

האם קיימת דרך לשמור קבוצה של נתונים בצורה שתקל עלינו את העבודה איתם?

אנו כותבים תוכניות מחשב על מנת לפתור מקרים שקורים בעולם האמיתי. לעיתים יש בעולם האמיתי בעיות שיכולות להיות מאפיינות בדרכים מסוימות. חישבו לדוגמה, איך לשמור במחשב תור של אנשים שמחכים לרופא בקופת חולים ואיך לשמור במחשב את עץ השורשים שלנו לעבודת השורשים בבית הספר.

הבעיות שתיארנו ממודלות באופן שונה, וכדי להכניס ולהוציא נתונים ביעילות אנו נדרשים לייצג ולשמור את הנתונים תחת מבני נתונים ספציפיים.

מסמך זה מציג את עיקרי המצגת המצורפת למשימה זו, המיועדת להעברה מול כיתת תלמידים. שימו לב כי המצגת מכילה הערות חשובות (מוטיבציה, נושאים לדיון ועוד).

מוטיבציה לפרק

בשיעור זה נלמד את הילדים על מבני הנתונים: מערך, תור, מחסנית ועץ בינארי. זהו בעינינו נושא חשוב ביותר מהסיבות הבאות:

- ניתן ללמידה ולהבנה על ידי ילדים.
- מפתח את החשיבה ואת ההבנה כיצד המחשב שלנו עובד.
- נושא חשוב ובסיסי במדעי המחשב.

במערך זה, ראשית נחדד בפני הילדים את העובדה שהמחשב שלנו צריך לשמור הרבה מאוד מידע, ולאחר מכן נסביר שהמידע הזה נשמר במבני נתונים מיוחדים. ננסה להדגיש בפני הילדים מהם ההבדלים בין מבני הנתונים השונים, ומהם היתרונות והחסרונות שלהם. במערך שיעור זה נשלב בין סוגים שונים של פעילויות בנושא: משחקי מחשב, דפי עבודה, פעילויות בסקרץ' ועוד, וזאת מתוך הניסיון שלנו שהילדים אוהבים גיוון בפעילויות.

הדגמה פתיחה לפני קבוצה:

מטרת המשימה: להבין את המשמעות של FIFO ו-LIFO.

עזרים נדרשים: חבילת קלפים.

אנחנו אומרים לילדים שכעת נראה להם קלפים ושהמטרה שלהם היא לזכור אותם בעל פה מהראשון לאחרון. מי שיזכור הכי הרבה קלפים, ינצח (לא חייבים לעשות את ההדגמה בצורת תחרות, אבל ממולץ).

אנחנו בוחרים מתנדב מהילדים ואומרים לו לערבב טוב את חפיסת הקלפים. לאחר מכן אנחנו מניחים את הקלפים בחפיסה הפוכה. אנחנו שולפים כל פעם קלף מראש החבילה ההפוכה, מראים אותו לתלמידים ושמים אותו הפוך בחבילה חדשה (שאנחנו יוצרים כעת). לאחר 7-10 קלפים אנחנו מפסיקים ומבקשים מהתלמידים להגיד מה היה הקלף הראשון ששלפנו, לאחר מכן מה הקלף השני וכו'... סביר להניח שאת הקלפים הראשונים הם יזכרו (כי ביקשנו מהם לזכור) אבל לא יותר מדי. כל פעם שמישהו זוכר את הקלף (צורה ומספר) אנחנו נותנים לו נקודה. המנצח הוא זה שקיבל את מספר הנקודות הגבוה ביותר בסיום.

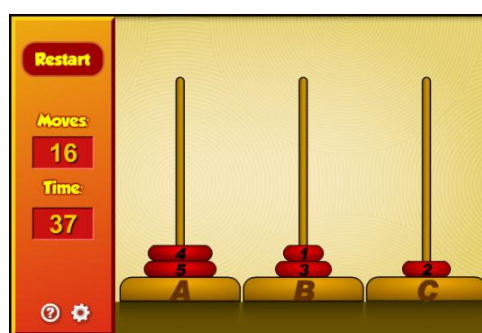
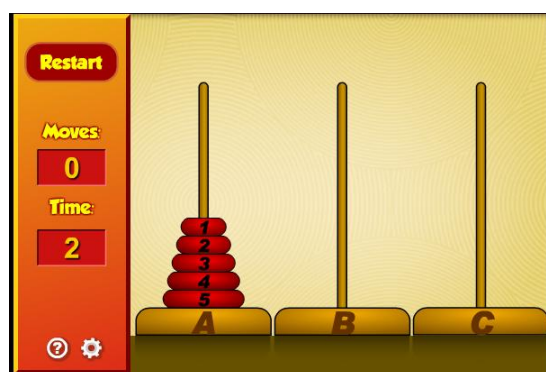
לאחר מכן אנחנו מבקשים מממתנדב נוסף לערבב שוב את הקלפים. כעת אנחנו אומרים לילדים שהם צריכים לזכור את הקלפים, אבל הפעם מהסוף להתחלה, כלומר מהקלף האחרון שאנחנו מראים להם, עד לקלף הראשון שהראינו להם.

אנחנו שוב נותנים להם לראות 10 קלפים, מוודאים שאף אחד לא רושם אותם (המשחק הוא לזכור מהזכרון). לאחר מכן אנחנו שואלים אותם מה הקלף האחרון שהראנו (סביר להניח שרובם יזכרו), לאחר מכן מה הקלף שלפניו, וכך הלאה.

משחק פתיחה

מגדלי האנו

המשחק: ישנן 5 דיסקיות על מוט A. אנו צריכים להעביר את כולן למוט C. ניתן להניח כל דיסקית רק על גבי דיסקית שגדולה ממנה.



לדוגמה המצב הזה הוא מצב אפשרי.

אנחנו נותנים לילדים לשחק במשחק. אפשר למצוא גרסת אינטרנט שלו בכתובת הבאה:

<http://www.novelgames.com/en/spgames/tower/>

אם אין אינטרנט פעיל אפשר לכתוב על הלוח את השלבים של המשחק.

מטרות המשחק הן:

- לתת לילדים לשחק קצת
- לתת להם התנסות בשמירת הדיסקיות על מוט (מה שמהווה מימוש של מחסנית, נושא בו נדון מאוחר יותר בשיעור).

דוגמאות לבסיסי נתונים

לאחר משחק הפתיחה נסביר לילדים מהו המושג שנקרא מבני נתונים. לשם כך, נביא את המשפט הממצה מויקפדיה ודוגמאות מוחשיות.

מתוך ויקיפדיה: "במדעי המחשב, מבנה נתונים הוא דרך לאחסון נתונים במחשב".

שאלה: במשחק מגדלי האנוי, אילו נתונים היינו צריכים לאחסן?

תשובה: את 5 הדיסקיות. תשובה טובה יותר: את 5 הדיסקיות ואת 3 המוטות.

דוגמאות חשובות למבני נתונים:

1. הכרת המושג מערך

- ניתן לתלמידים את הבעיה הבאה: נניח שאנחנו רוצים ליצור תוכנית ששומרת ציונים של 20 תלמידים ומחשבת ממוצע אחד של כולם. שאלה: איך נעשה זאת? תשובה אפשרית: נשמור 20 משתנים.
- אם התלמידים לא מעלים את התשובה הזו, נעלה אותו בעצמנו: "אפשר לשבת ולהגדיר 20 משתנים חדשים כל פעם (כמו בסקראצ')".
- נציג את הבעיה עם פתרון זה - נצטרך לבחור 20 פעם ב"משתנה חדש" ולבחור לכל אחד שם שונה. זה יקח הרבה זמן ובנוסף גם יהיה מסורבל ומבולגן כי יהיו לנו 20 משתנים פזורים על המסך.
- **נושא לדיון:** נדון עם הילדים האם הפתרון הנ"ל מספיק טוב, ונסה לגרום להם להבין שזהו פתרון בעייתי מבחינות שונות:
 1. זמן
 2. סרבול
 3. לא ריאלי עבור כמות גדולה של תלמידים
 4. לא יעיל מבחינת זיכרון
- נאמר לתלמידים שהיה עדיף אם היינו יכולים להגדיר מבנה נתונים אחד גדול שמכיל בתוכו מקום ל 20 מספרים (הציונים של התלמידים).
- נסביר להם שמבנה שמכיל מספר קבוע של משתנים, נקרא מערך

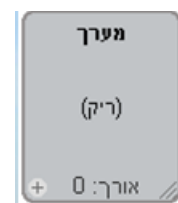
פעילות בסיסית בסקראצ' בנושא מערך

הערה: מאחר ומערך השיעור מועבר ברובו בכיתה (ולא בחדר המחשבים), ניתן לדלג בשלב זה על פעילויות הסקרץ' ולהעביר אותן בשיעור הבא (כך אנו עשויים כשהעברנו את מערך השיעור).

בעולם האמיתי, שפות התכנות שבהן משתמשים מתכנתים מאפשרות להם להגדיר מבני נתונים שונים. גם בשפת סקרץ' ניתן להגדיר מבני נתונים. נלמד כעת כיצד ליצור מערכים בסקרץ' וכיצד להשתמש בהם.

בלשונית משתנים, נבחר ב"צור רשימה". נקרא לרשימה זו "מערך".

כעת סיימנו ליצור מערך ריק:



- ניתן להוסיף פריטים למערך
- ניתן למחוק פריטים מהמערך
- ניתן להחליף פריט במערך עם פריט אחר
- ניתן לקבל את מספר הפריטים במערך

המשימה (דורשת ידע בסיסי קודם בשפת סקריץ'):

- צור מערך בסקריץ' בגודל 10.
- מלא את המערך במספרים אקראיים מ-1 ועד 100.
- חשב כמה מספרים זוגיים קיימים במערך שיצרת.



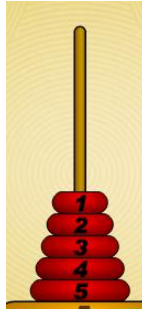
- אם, לדוגמא, זהו המערך שלנו:
- אז נחזיר 2, כי במערך הזה יש שלושה מספרים ורק שניים מהם זוגיים.

2. דוגמאות למבני נתונים תור ומחסנית

לפעמים אנו כותבים תוכנית מחשב שנדרשת למבנה נתונים בעל תכונות מסוימות.

לדוגמה, אם היינו רוצים לשמור את הדיסקיות על המוט במגדלי האנוני היה מאוד עוזר לנו אם המוט היה בעל התכונות הבאות:

1. אפשר לשים עליו דיסקית (חוקית, שקטנה מהדיסקית שבראש המוט).
2. בכל שליפה היה אפשר להוציא מהמוט רק את הדיסקית האחרונה.



- מבנה נתונים שהאיבר האחרון שנכנס אליו הוא האיבר הראשון שיוצא קוראים מבנה שעובד בשיטת **LIFO**. המשמעות של ראשי התיבות באנגלית הם: *Last in first out*. כלומר האיבר האחרון שנכנס למבנה – הוא יהיה הראשון לצאת.
- מבנה נתונים שעובד בצורת **FIFO** – *First in first out*, הוא מבנה נתונים שעובד בצורה בה האיבר הראשון שנכנס, הוא יהיה הראשון לצאת.
- נראה שתי דוגמאות למבני נתונים שעובדים בשיטות אלו שנקראים: "**מחסנית**" ו"**תור**".

מבנה הנתונים תור

נניח שאנחנו רוצים ליצור תוכנה שעוזרת לנהל סניף של קופת חולים. בקופת החולים אנשים באים לקבל שירות מרופא. כל אדם שמגיע לקופת חולים לוקח מספר, ואז מחכה על ספסל עד שמגיע תורו. כשמגיע תורו הוא נכנס לרופא והתור מתקדם. אנחנו רוצים לעשות תוכנה שמנהלת את האנשים בקופת חולים, כלומר מוסיפה לסוף התור כל חולה חדש שנכנס, וכל פעם שהרופא מסיים הוא קורא לחולה שהתור שלו הגיע. כשהחולה נכנס לחדר הרופא, הרופא מקדם את התור.

נבקש מהתלמידים להציע רעיונות לניהול התור: לשמור ולהוציא כל פעם את האדם הרלוונטי בתור.

נדריך את התלמידים שמה שבעצם חשוב לנו זה 3 פעולות:

- הכנסת אדם חדש לתור.
- בדיקה מי כרגע הראשון בתור.
- הוצאת האדם שבראש התור (קידום התור באחד).

שאלות חשיבה נוספות לתלמידים:

מה עוד חשוב לנו לעשות בתור? לדוגמה, האם צריך לממש פעולה של הוצאת אדם מאמצע התור? תשובה: לא כי אנחנו מניחים שכל בן אדם יצא מהתור רק לאחר שיגיע תורו (כלומר לאחר שיתקדם לראש התור) ולכן פעולה זו מיותרת.

מבנה הנתונים מחסנית

כדי להציג את מבנה נתונים זה, נציג לילדים את הדוגמה הבאה:

"קודם" כפתור

נניח שאנחנו רוצים ליצור דפדפן (כמו כרום או פיירפוקס). כחלק מיצירת הדפדפן אנחנו צריכים ליצור כפתור "הקודם". כל פעם שנלחץ על הכפתור הזה הדפדפן שלנו יציג את הדף הקודם שהיינו בו (זהו דף האינטרנט שבו היינו לפני שנכנסו לדף שאנחנו נמצאים בו עכשיו).



כדי שכולם יבינו מה עושה הכפתור "הקודם" נעשה דוגמה:

נניח שהתחלנו את הגלישה בגוגל, אחר כך גלשנו לפייסבוק, אחר כך ליוטוב, ואחר כך ל YNET.

אם נלחץ כעת על לחצן "הקודם" נגיע ליוטוב (האתר הקודם שהיינו בו).

מה יקרה אם נלחץ לאחר מכן על "הקודם"? ואחר כך?

נבהיר לילדים שלמרות שיש בעיתיות מסוימת אם מסתכלים על הכפתור כחלק מדפדפן אמיתי כיוון שבו אנו יכולים ללכת גם קדימה, או למקום ספציפי בהיסטוריה (בניגוד להגדרת המחסנית), ולכן אנו מסתכלים אך ורק על הכפתור "הקודם" ככפתור בפני עצמו ולא כחלק מדפדפן.

לאחר שכולם יבינו את משמעות כפתור "הקודם", נשאל מה אנו צריכים לשמור במבנה נתונים ליצירת הכפתור? תשובה: את כל דפי האינטרנט הקודמים שהיינו בהם.

נשאר את תלמידים האם **האם תור תהיה בחירה טובה?** אם לא, למה?

תשובה: **תור** אינו בחירה טובה, מכיוון שהתור נותן לנו את האתר **הראשון** שנשמר ואילו אנחנו צריכים להוציא את האתר **האחרון** שנשמר.

נשאל את התלמידים איך היה כדאי לשמור את הנתונים.

נציג את המבנה מחסנית:

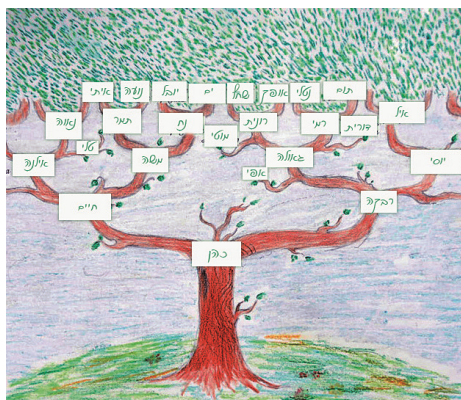
מבנה זה שומר (כמו במחסנית אמיתית ואף אפשר להראות תמונה של מחסנית אמיתית) את הפריטים בו לפי סדר הגעתם כשהפריט האחרון שהגיע יהיה הראשון לצאת. במקרה של כפתור "הקודם" בדפדפן שלנו, המחסנית תשלוק לנו בכל לחיצה את השם של האתר **האחרון** אליו נכנסו.



נשאל את התלמידים אם ראינו היום בשיעור עוד דוגמה שיכולה להשתמש במבנה מחסנית. תשובה: **כן, המוט של מגדלי האנוני הוא בעצם מבנה נתונים שצריך להשתמש במבנה נתונים מסוג מחסנית.**

מבנה הנתונים – עץ בינארי

כרגיל, נפתח בדוגמה מהחיים האמיתיים. נניח ובמסגרת עבודת השורשים, המורה ביקשה מאיתנו להכין אילן יוחסין. העץ שנבנה יראה בערך כך (תלוי כמה גדולה המשפחה שלכם):



ניתן לראות שבראש משפחת כהן נמצאים רבקה וחיים.

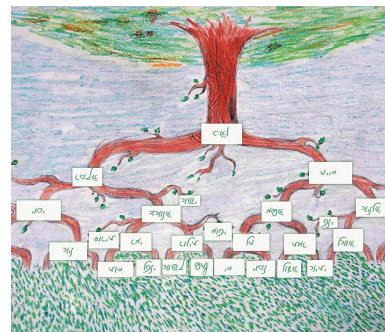
לחיים שני ילדים- משה ואילנה

לרבקה שלושה ילדים- יוסי גאולה ואפי.

וכך הלאה...

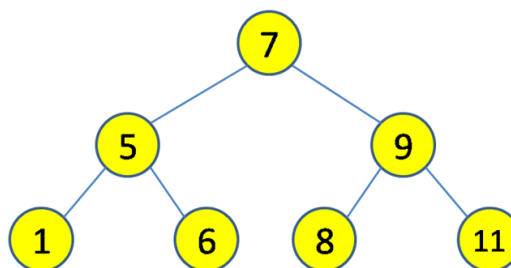
כעת נסתכל כעת על העץ בצורה הפוכה:

לעץ יש שורש, לשורש שני ילדים, לכל אחד מהילדים ילדים משלו, וכך הלאה...



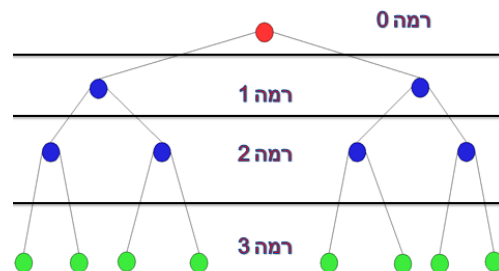
נסביר לילדים שבמדעי המחשב, עץ הוא מבנה נתונים אשר מדמה היררכיה של עץ – לעץ יש שורש ותתי עצים של ילדים.

עץ בינארי הוא עץ שבו לכל צומת יש 2 ילדים לכל היותר.



נשאל את הילדים האם 9 הוא הבן של 7 או 11, ומי האבא של 6. מטרת השאלות: לוודא שהם באמת הבינו את המושגים הבסיסיים.

כעת, נסביר שבעץ בינארי קיים המושג "רמה".



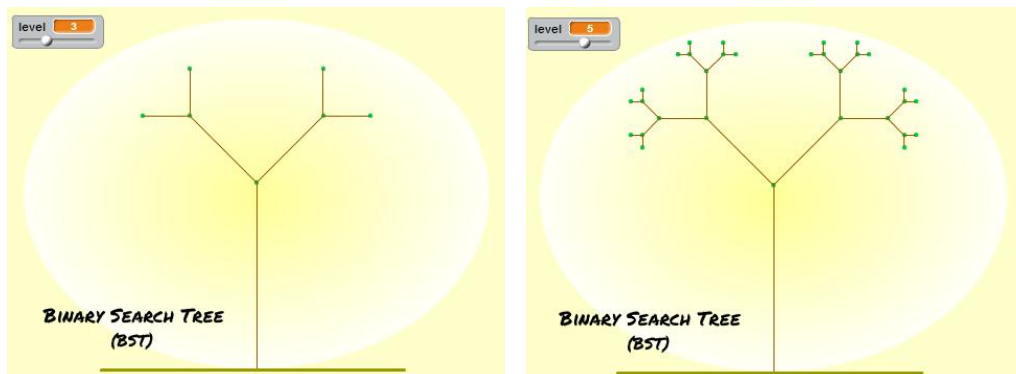
נשאל את התלמידים כמה צמתים יש בכל רמה. ננסה לגרום להם להבין שבכל רמה מספר הצמתים הוא 2 בחזקת מספר הרמה.

דוגמא בסקרן

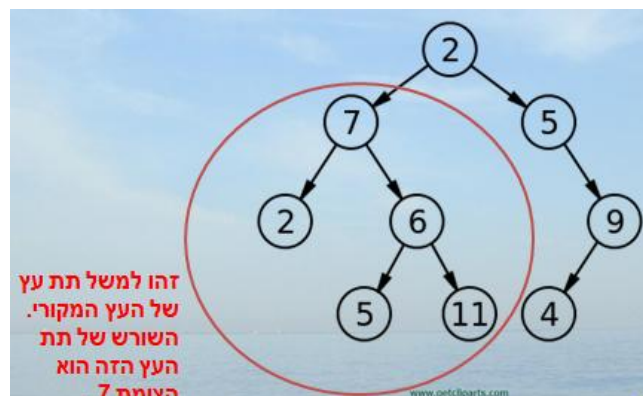
במידה והילדים נמצאים מול מחשב, נבקש מהם להיכנס לכתובת:

<http://scratch.mit.edu/projects/13100004/#editor>

זהו מימוש של עץ בינארי, אשר מצייר את העץ המבוקש בהתאם לרמה שמזינים. ניתן לילדים לקח עם זה במטרה שהמושג "רמה של עץ" יהיה כר ברור יותר.

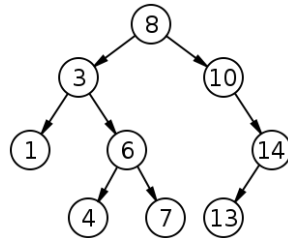


נסביר קצת על המושג "תת עץ": תת עץ הוא כל עץ שניתן ליצור מצומת כלשהו של העץ המקורי, הכולל את הצומת עצמו ואת כל הצאצאים שלו (ילדיו, נכדיו וכו')



נסביר מהו עץ חיפוש - במדעי המחשב עץ חיפוש הוא מבנה נתונים ממוין המאפשר הכנסה, הוצאה וחיפוש מהירים. הכלל הוא כזה: כל צומת חייב להיות גדול מהבן השמאלי שלו ומכל צאצאיו, וקטן מהבן הימני שלו ומכל צאצאיו.

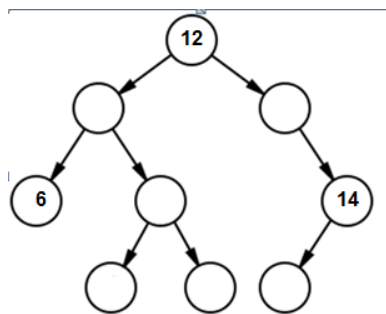
דוגמה לעץ חיפוש:



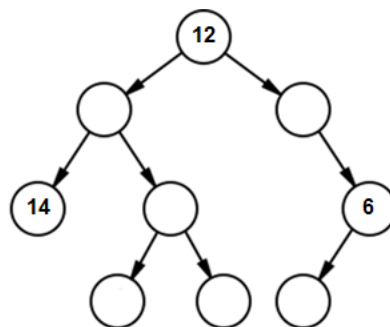
השורש, שהוא 8, גדול מהבן השמאלי שלו – 3 ומכל ילדיו (הצמתים 1,3,6,4,7), וקטן מהבן הימני שלו – 10 ומכל ילדיו (10,14,13)

פעילות כיתתית משותפת:

נסו למלא את הצמתים הריקים במספרים שימלאו את הכלל שלמדנו



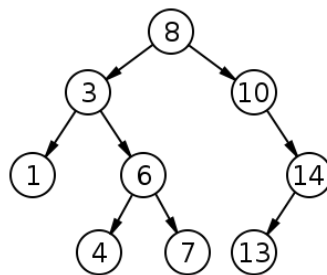
שאלת מחשבה: האם ניתן למלא את העץ הבא לפי הכלל שהגדרנו?



התשובה היא כמובן שלא, למשל כי 6 אינו גדול מ-12, או כי 14 אינו קטן מ-12.

טיול על העץ

בעזרת אלגוריתם נכון, ניתן לטייל על העץ וכך למעשה לסרוק את כל הצמתים שלו. הטיול יתחיל תמיד משורש העץ.



דוגמאות לטיול על העץ: מתחילים בצומת 8, משם פונים ימינה לצומת 10, משם לצומת 14 ומסיימים את הטיול בצומת 13.

דף עבודה מספר 1 בנושא מבני נתונים

לפניך מתוארת רשימת מצבים הדורשים שימוש במבנה נתונים. רשום ליד כל מצב באיזה מבנה נתונים מבין אלו שלמדנו (מערך, תור, מחסנית או עץ) היית ממליץ להשתמש.

1. אנחנו יוצרים כפתור "בטל פעולה אחרונה" לתוכנת ציור (כמו "הצייר" או "פוטושופ"). כשלוחצים על הכפתור הזה, תוכנת הציור חוזרת לתמונה שהייתה לפני השינוי האחרון שעשינו. לדוגמה אם ציירנו בלון שקוף, צבענו אותו לאדום, ואז התחרטנו ולחצנו על כפתור "בטל פעולה אחרונה", התמונה שנקבל תהיה של בלון שקוף (כמו שהייתה לפני הפעולה האחרונה – לפני שצבענו באדום את הבלון).
מבנה הנתונים הנדרש הוא: _____.

2. הסטודנטים שואלים שאלה בכיתה. הרבה תלמידים מרימים את היד, אבל רק אחד יכול לדבר. הסטודנטים רוצים לתת לכל מי שהצביע לדבר, אבל לא ביחד, אלא כל אחד לחוד.
מבנה הנתונים הנדרש הוא: _____.

3. אנחנו צריכים לשמור 100 מספרי טלפון של ילדים בבית ספר.
מבנה הנתונים הנדרש הוא: _____.

4. במפעל מסוים במרכז הארץ ישנם עובדים חדשים וותיקים. כשמתקבלת עבודה חדשה במפעל, נותנים אותה לעובד החדש ביותר (כי הוותיקים כבר מבוגרים ועשו הרבה במפעל, והצעירים רק הגיעו).
מבנה הנתונים הנדרש הוא: _____.

5. רוצים לשמור במחשב ספר טלפונים של 2000 אנשים. כל רשומה בספר הטלפונים תכיל שם וטלפון.

מבנה הנתונים הנדרש לשמירת השמות הוא:

_____.

מבנה הנתונים הנדרש לשמירת הטלפונים הוא:

_____.

משימה למתקדמים: נסו לחשוב בעצמכם על מקרה בחיים האמיתיים שמבני הנתונים שלמדתם מצריך וכתבו אותו למטה. לאחר מכן תלמיד אחר ינסה לגלות את המבנה.

6. תיאור מבנה הנתונים:

שם: _____

דף עבודה מספר 2 בנושא מחסנית

אנחנו משתמשים במחסנית מסוימת על מנת לשמור מספרים. יש לנו 3 פעולות:

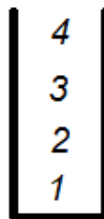
- **הכנס מספר למחסנית** – הכנסת מספר לראש המחסנית.
- **הוצא מספר מהמחסנית** – הוצאת מספר מראש המחסנית.
- **הצץ למחסנית** – בדיקה מהו המספר שנמצא בראש המחסנית.

הכנס למחסנית (17) – תכניס את המספר 17 לראש המחסנית.

הוצא מהמחסנית – מוציאה החוצה את האיבר שבראש המחסנית.

הצץ למחסנית – אומרת לנו איזה מספר יש בראש המחסנית כרגע, אבל לא מוציאה אותו.

דוגמה: אם התחלנו עם מחסנית ריקה ואז ביצענו את הפעולות: הכנס למחסנית (1), הכנס למחסנית (2), הכנס למחסנית (3), הכנס למחסנית (4), הכנס למחסנית (5), הוצא מהמחסנית. בסוף הפעולות המחסנית תראה ככה:



כי הכנסנו קודם את 1, אח"כ את 2, 3, 4, 5, ואז הוצאנו את האיבר האחרון (שהיה 5).

אנחנו מתחילים עם מחסנית ריקה ואז עושים את הפעולות:

הכנס למחסנית (2), הכנס למחסנית (4), הכנס למחסנית (6), הוצא מהמחסנית ,
הוצא מהמחסנית , הכנס למחסנית (12), הכנס למחסנית (55), הכנס למחסנית (2),
הוצא מהמחסנית, הצץ למחסנית, הוצא מהמחסנית, הכנס למחסנית (100).

צייר את המחסנית שנקבל בסוף כל הפעולות הללו:



דף עבודה מספר 3 בנושא מחסנית ותור

משימה מספר 1

בדף זה ניצור את מבנה הנתונים מחסנית בסקרץ'.

בסיום דף עבודה זה נקבל את התוכנית הבאה:

איזה פעולה ברצונך לבצע במחסנית?

מחסנית	
1	1
2	2

אורך: 2 +

✓

התכנית שלנו עובדת בצורה הבאה: הילד שואל ללא הפסקה איזה פעולה ברצוננו לבצע במחסנית.

אם נכתוב "הכנסה": כאן נממש פעולת הכנסה למחסנית. הילד ישאל איזה מספר ברצוננו להכניס למחסנית, ומספר זה יוכנס לראש המחסנית.

אם נכתוב "הוצאה": כאן נממש פעולת הוצאה מהמחסנית.

אם נכתוב "בדיקה": הילד יאמר מהו המספר שנמצא כעת בראש המחסנית.

בסיום דף עבודה זה תוכל לבדוק את התשובה שלך לדף העבודה הקודם.

משימה מספר 2

שנה את התכנית שכתבת כך שמבנה הנתונים יהפוך ממחסנית לתור:

- פעולת "הכנסה" תכניס מספר לסוף התור
- פעולת "הוצאה" תוציא מספר מראש התור
- פעולת "בדיקה" תחזיר את המספר שבראש התור

דף עבודה מספר 4 בנושא עצים בינאריים

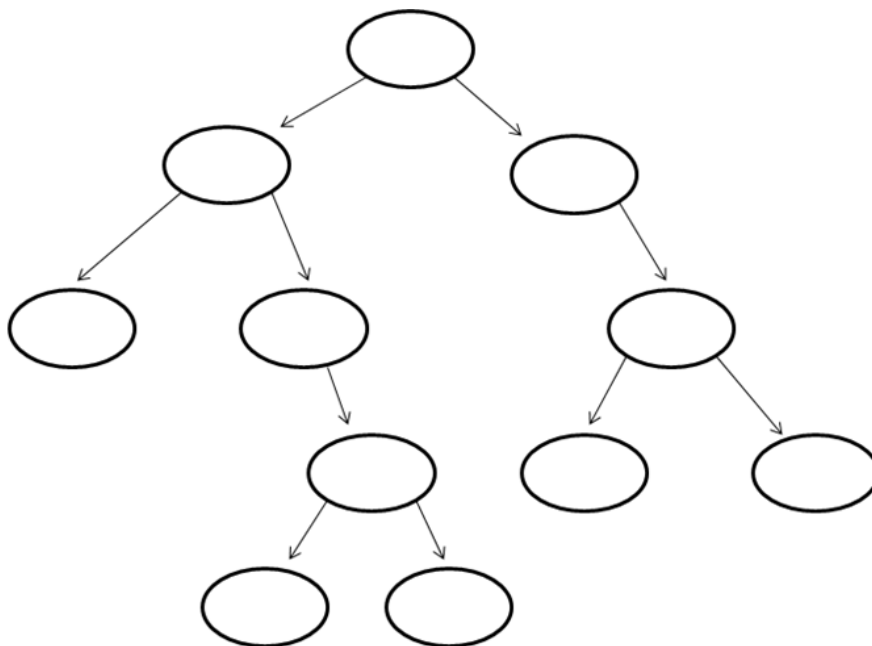
משימה מספר 1

הכירו את משפחת לוי.

לאבא פנחס יש שני ילדים- דניאלה ושרה.

לדניאלה יש ילד אחד ושמו ברק ולברק יש שני ילדים- רותי וזיו. לשרה יש שני ילדים- קובי וקרן. לקרן יש ילד אחד – רפי, ולרפי שני ילדים- דבי ורינה. לקובי אין ילדים.

מלא את אילן היוחסין הבא של משפחת לוי.



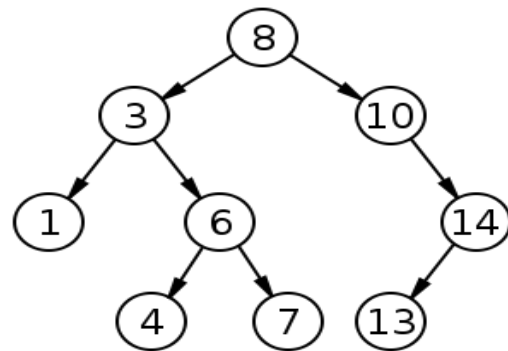
נרצה להפוך כעת את העץ לעץ חיפוש בינארי.

פנחס הציע שנשתמש באילן היוחסין של משפחת לוי באופן הבא: נחליף כל שם בגיל. אם למשל פנחס הוא בן 72, נחליף את השם של הצומת "פנחס", בגילו של פנחס-72.

פנחס טועה- אם נשתמש בפתרון שלו, לא נקבל עץ חיפוש בינארי.

הסבר מדוע:

משימה מספר 2



נתון עץ חיפוש בינארי.

נרצה לכתוב אלגוריתם לחיפוש מספר כלשהו בעץ.

השלם את האלגוריתם:

בהינתן מספר כלשהו לחיפוש x ,

1. התחל משורש העץ.

2. אם $x =$ שורש העץ, האלגוריתם הסתיים.

3. אם $x <$ שורש העץ לך לתת העץ הימני/השמאלי (הקף בעיגול את התשובה הנכונה).

4. אם $x >$ שורש העץ לך לתת העץ הימני/ השמאלי (הקף בעיגול את התשובה הנכונה).

5. חזור לשלב ____ של האלגוריתם.