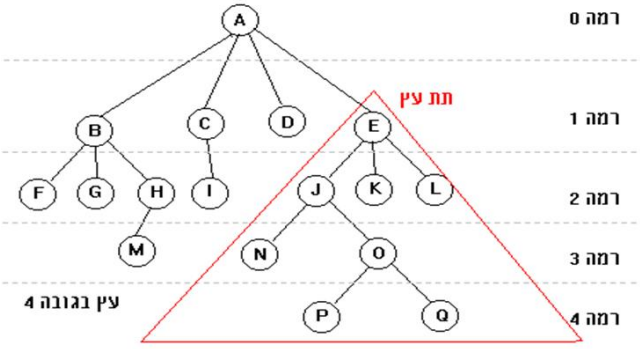
עצים בינאריים – BinTreeNode

עץ - מבנה נתונים (גרף) המורכב מצמתים המקיימים ביניהם יחס של הורה ובנים / צאצאים.

צומת - חוליה ולה תכונות: value - הפניה לטיפוס המידע, והפניות לצמתים אחרים.
קשת - הפניה היוצאת מצומת **אב** לצומת **בן**.



כל עץ מורכב מצומת **שורש** (**root**) שהוא הצומת הראשון בעץ.
לכל צומת פרט לשורש יש **הורה / אב** (perent).
כל צומת יכול להיות הורה לצמתים רבים אחרים,
וכל הצמתים הללו הם **בניו** של ההורה.

כל הצמתים שהם בניו של אותו הורה הם צמתים **אחים**.

כל צומת בעץ הוא **צאצא** (descendent) של כל הצמתים שמעליו.
כל צומת הוא **אב קדמון** של כל צאצאיו.

מספר בניו של הצומת קובעים את דרגתו - **דרגת צומת** (degree).
לצומת שלו 3 בנים נקבעת דרגה 3 ואילו לצומת שאין לו בנים כלל נקבעת דרגה 0.

צומת שדרגתו 0 קרוי **עלה** (**leaf**).

עץ בינארי (BinNode) - עץ שבו הדרגה המקסימאלית של כל אחד מצמתיו הוא 2. כלומר - לכל צומת יש שתי הפניות left ו- right לשני בנים

מסלול - רצף של צמתים/קשתות היוצאות מצומת אחת עד צומת שני.

רמת עץ (tree level), אורך המסלול המקסימלי משורש העץ ועד העלה. השורש בעץ מסומן כרמה 0, בניו של השורש נמצאים ברמה מספר 1, בני בניו ברמה 2 וכך הלאה.
רמה של צומת הינה מספר הקשתות שיש לעבור כדי להגיע משורש העץ עד הצומת המבוקש.

גובה עץ או עומק העץ הוא מספרה של הרמה הגבוהה ביותר בעץ.
מספר הקשתות לאורכו של המסלול הארוך ביותר היוצא מן השורש ומסתיים בעלה.

בן-שמאלי (**תת-עץ-שמאלי**) - עץ בינארי ששורשו הוא הבן השמאלי של צומת נתון.
בן-ימני (**תת-עץ-ימני**) - עץ בינארי ששורשו הוא הבן הימני של צומת נתון.

בן הוא תמיד שורש של **תת-עץ** (sub-tree).

רמה מלאה - רמה בעץ שקיימים בה כל הצמתים.

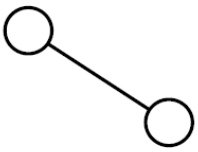
עץ בינארי מלא (**עץ שלם**) - עץ שכל רמה בו היא רמה-מלאה.

עץ בינארי כמעט מלא - עץ בינארי שכל הרמות פרט לרמה האחרונה מלאות, וכל הצמתים ברמה האחרונה מרוכזים בצד שמאל.

עץ בינארי לחלוטין (עץ לא בהכרח מלא) - עץ שדרגת כל צומת בו היא 0 או 2
(כלומר - אין צומת שלו בן יחיד).

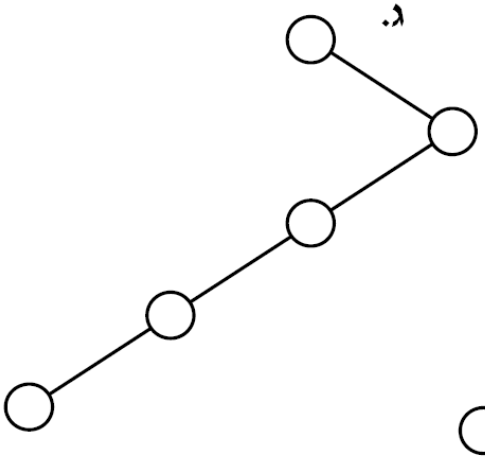
דוגמאות לעצים בינריים:

א.

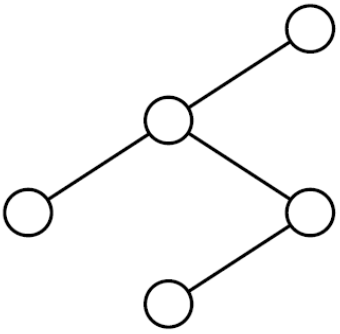


ב.

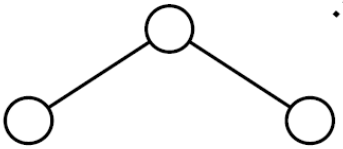
ג.



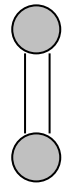
ד.

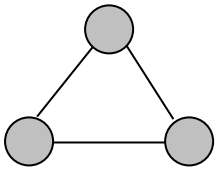


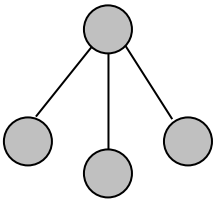
ה.

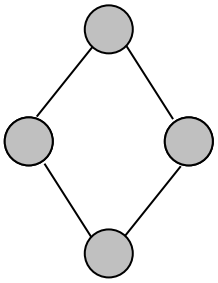


דוגמאות לעצים שאינם עצים בינאריים:



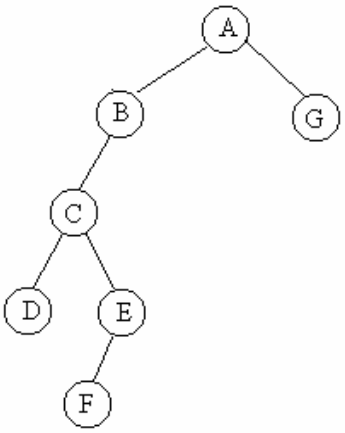






תרגילים:

1. אלו מהמשפטים הבאים נכונים לגבי העץ שלהלן? הסבר:



א. A הוא אב קדמון של כל צמתי העץ האחרים. ----------

ב. B הוא אב קדמון של G. ----------

ג. F הוא צאצא של C, E ו-G. ----------

ד. D הוא ברמה 2. ----------

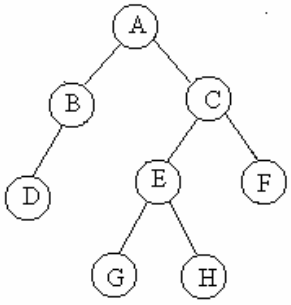
ה. D הוא אח של E. ----------

ו. C הוא אב קדמון של F ו-D. ----------

ז. רמתם של אחים בעץ זהה. ----------

---

2. כמה אבות קדמונים יש לצומת ברמה n בעץ בינארי ?

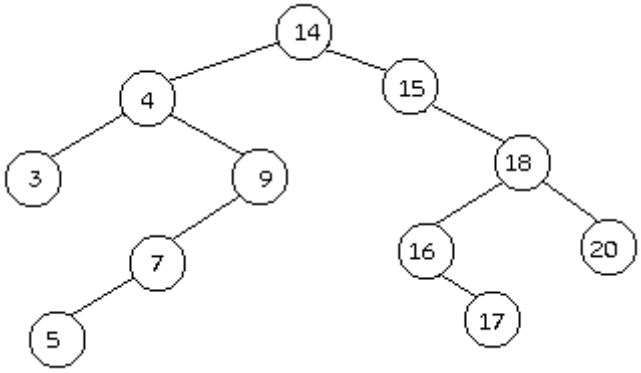


---

3. מהו מספר הצמתים ברמה n בעץ בינארי מלא ?

4. מהו מספר הצמתים הכולל בעץ בינארי מלא שרמתו n ?

סריקת עץ בינארי:



סריקה בסדר תחילי - pre order

1. בקר בשורש.
2. סרוק תת-עץ השמאלי בסדר תחילי.
3. סרוק תת-עץ ימני בסדר תחילי.

סריקה בסדר תוכי – in order

1. סרוק תת-עץ השמאלי בסדר תוכי.
2. בקר בשורש.
3. סרוק תת-עץ ימני בסדר תוכי.

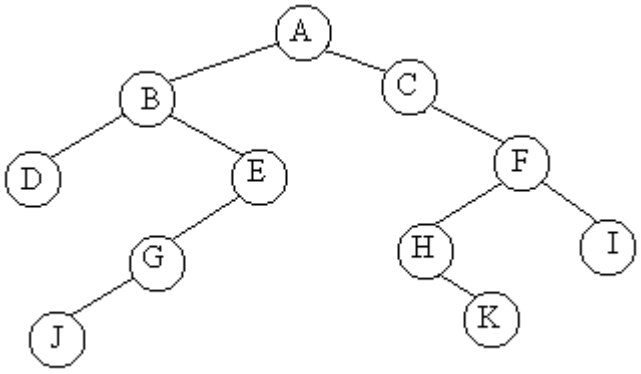
סריקה בסדר סופי - post order

1. סרוק תת-עץ השמאלי בסדר סופי.
2. סרוק תת-עץ ימני בסדר סופי.
3. בקר בשורש.

שלושת שיטות הסריקה מעידות על מיקומה של פעולת הביקור בשורש בין 3 הפעולות.

קיימת שיטת סריקה נוספת - סריקה לפי רמות.

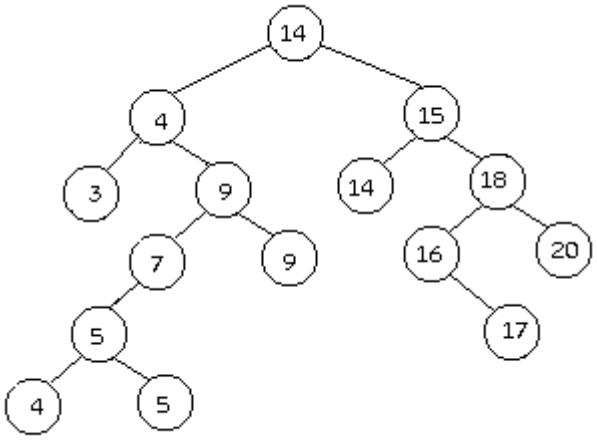
- שחזור עץ בינארי על פי 2 מסריקותיו.



סריקת העץ בשיטות השונות:

סריקה בסדר תחילי:	A B D E G J C F H K I	PreOrder
סריקה בסדר תוכי:	D B J G E A C H K F I	InOrder
סריקה בסדר סופי:	D J G E B K H I F C A	PostOrder

תרגיל: מהו סדר הביקור בצמתים בכל אחת משלוש שיטות הסריקה?



in order: _____

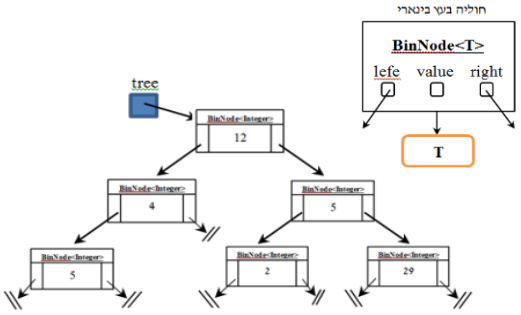
pre order: _____

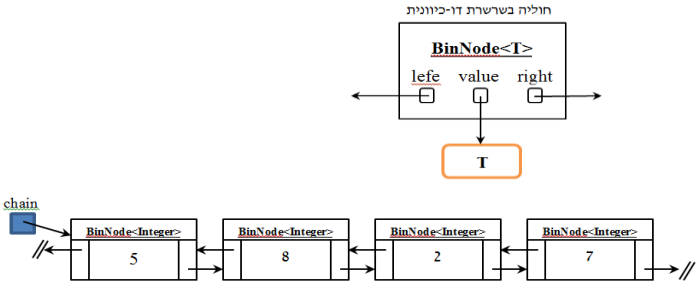
post order:_____

חוליה בינארית BinNode<T>

חוליה המכילה ערך מטיפוס Tוהפנייה לשתי חוליות בינאריות.
ניתן להשתמש במחלקה זאת כדי לייצג עץ בינארי או שרשרת דו-כיוונית.

תיאור הפעולה	חתימת הפעולה	סיבוכיות
בנאי:		
פעולה הבונה חוליה בינארית שערכה x ושתי ההפניות null	BinNode (T x)	O(1)
פעולה הבונה חוליה בינארית שערכה x, ושתי ההפניות שלה הן left ו- right (ערכם של הפרמטרים left ו- right יכול להיות null)	BinNode (BinNode<T> left, T x, BinNode<T> right)	O(1)
שאילתות:		
אם T מחלקה עוטפת לטיפוס בסיסי (Integer, Double, Character) יוחזר ערך החוליה, ואם הפניה לעצם, תוחזר הפנייה לעצם זה	T GetValue ()	O(1)
פעולה המחזירה הפנייה לחוליה אליה מפנה left.	BinNode <T> GetLeft ()	O(1)
פעולה המחזירה הפנייה לחוליה אליה מפנה right.	BinNode <T> GetRight ()	O(1)
פעולה המחזירה אמת אם יש חוליה משמאל, ושקר אחרת	**boolean** HasLeft ()	O(1)
פעולה המחזירה אמת אם יש חוליה מימין, ושקר אחרת	**boolean** HasRight ()	O(1)
פעולה המחזירה את מצב החוליה כמחרוזת. (*) סיבוכיות: אם T עצם מטיפוס פשוט ← O(1) ואם T מייצג אוסף באורך כלשהו ← O(|T|)	String ToString ()	O(|T|) (*)
פקודות:		
הפעולה משנה (מעדכנת) את ערך החוליה ל- x	**void** SetValue (T x)	O(1)
הפעולה משנה את ערכה של ההפניה left ל- left הפרמטר left יכול להיות גם null	**void** SetLeft (BinNode <T> left)	O(1)
הפעולה משנה את ערכה של ההפניה right ל- right הפרמטר right יכול להיות גם null	**void** setRight (BinNode <T> right)	O(1)





תכנית לבניית עץ בינארי

עקבו אחר קוד המחלקה וציירו את העץ המתקבל:

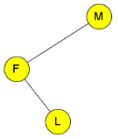
```
public class PrintTree
{
	public static void Main(string[] args)
	{
		BinNode<char> t1, t2, bt1, bt2, bt;

		//--- ……. ---
		t1 = new BinNode <char>('K');
		t2 = new BinNode <char>('M');
		t2.SetLeft(new BinNode<char>(null,'F',new BinNode <char> ('L')));

		bt1 = new BinNode<char>(t1,'B',t2);
		bt1 = new BinNode<char>(null,'J', bt1);

		//--- ……. ---
		t1 = new BinNode<char>('D');
		t2 = new BinNode<char>('C');

		t2.SetLeft(new BinNode<char>('I'));
		t2.SetRight(new BinNode<char>(new BinNode<char>('G'),'H',null));

		bt2 = new BinNode<char>(t1,'E',t2);
		bt = new BinNode<char>(bt1,'A',bt2);

	}
}
```

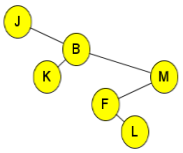
א. צייר את העץ המתקבל
ב. העתק את התכנית למחשב ובדוק את נכונות השרטוט שהתקבל בסעיף א'.

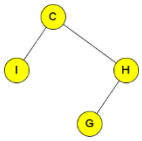
תכנית לבניית עץ בינארי

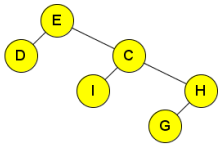
העץ המתקבל:

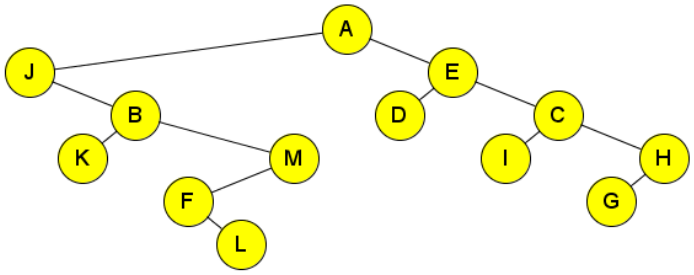
```
public class PrintTree
{
	public static void Main(string[] args)
	{
		BinNode<char> t1, t2, bt1, bt2, bt;

		//---..................................... ---
		t1 = new BinNode <char>('K');
		t2 = new BinNode <char>('M');

		t2.SetLeft(new BinNode<char>(null,'F',new BinNode <char> ('L')));

		bt1 = new BinNode<char>(t1,'B',t2);
		bt1 = new BinNode<char>(null,'J', bt1);

		//--- ..................................... ---
		t1 = new BinNode<char>('D');
		t2 = new BinNode<char>('C');

		//--- ..................................... ---
		t2.SetLeft(new BinNode<char>('I'));
		t2.SetRight(new BinNode<char>(new BinNode<char>('G'),'H',null));


		bt2 = new BinNode<char>(t1,'E',t2);
		bt = new BinNode<char>(bt1,'A',bt2);
	}
}
```











אוסף פעולות לעץ בינארי:

- מספר הצמתים בעץ
- סכום הצמתים בעץ

- עלה?
- מספר העלים בעץ
- סכום העלים
- מספר הצמתים שערכם x
- מספר צמתים זוגיים

- מספר בנים ימניים בעץ
- מספר בנים יחידים בעץ
- מספר הבנים של צומת נתון
- מספר הנכדים של צומת נתון
- עץ פרו ורבו (2004) - מחזיר אמת אם קיים צומת ולו לפחות שני נכדים משני הבנים.
- מספר-סבים בעץ

- האם-נמצא? (bt, x)
- האם כל הצמתים אי-זוגיים?
- עץ-ממוצע? - אמת אם ערך כל צומת שווה לממוצע הערכים של בניו.
- ממוין-רמות? - אמת אם ערך כל אב קטן מערך בניו, ושקר אחרת

- סריקה לפי רמות
- רמת-צומת (bt, t) - t הפניה לצומת בעץ bt
- סכום צמתים ברמה (bt, level)
- מספר צמתים ברמה (bt, level)
- האם אחים? (bt, t1, t2)

- גובה-עץ
- ערך מקסימאלי בעץ
- סכום צמתים ברמה
- מספר צמתים ברמה

- האם-אחים? (bt1, bt2)
- האם-הורה? (bt1, bt2)

- סבא רבא (bt, tX, tY) - בגרות - מחזיר הפניה לצומת שהוא אב-קדמון הכי קרוב לשני עלים נתונים.
- צאצא? (t1, t2) מחזיר אמת אם הצומת t1 הוא צאצא של t2 ושקר אחרת
- הורה (bt, son) - מחזיר הפניה לצומת ההורה של הצומת son. הנחה: $bt \neq son$
- מסלול-אחיד - האם קיים מסלול מצומת השורש לאחד העלים שערכי כל צמתיו זהה.
- מספר צמתים ברמה k
- עץ-מלא? - כל הרמות מלאות (הקשר שבין מספר הצמתים N וגובה העץ K: $2^k = n$)
- **עץ בינארי לחלוטין** (עץ מלא) - עץ שדרגת כל צומת בו היא 0 או 2
- בגרות 2007 - מסלול אחיד (בגרות 1996 - עץ בעל שרשרת x)

- עץ ביטוי
- **עץ חיפוש בינארי**
- עץ ממומש במערך (1999)

- עץ שה- value שלו הוא מבנה נתונים מסוג: מערך / רשימה / מחסנית / תור / עץ
- ייצוג שונה לעץ בינארי (באמצעות מערך)
- עץ טרינארי