Lecture3 of Geiger & Itai's slide brochure
www.cs.technion.ac.il/~dang/courseDS

עצים ועצי חיפוש

חומר קריאה לשיעור זה

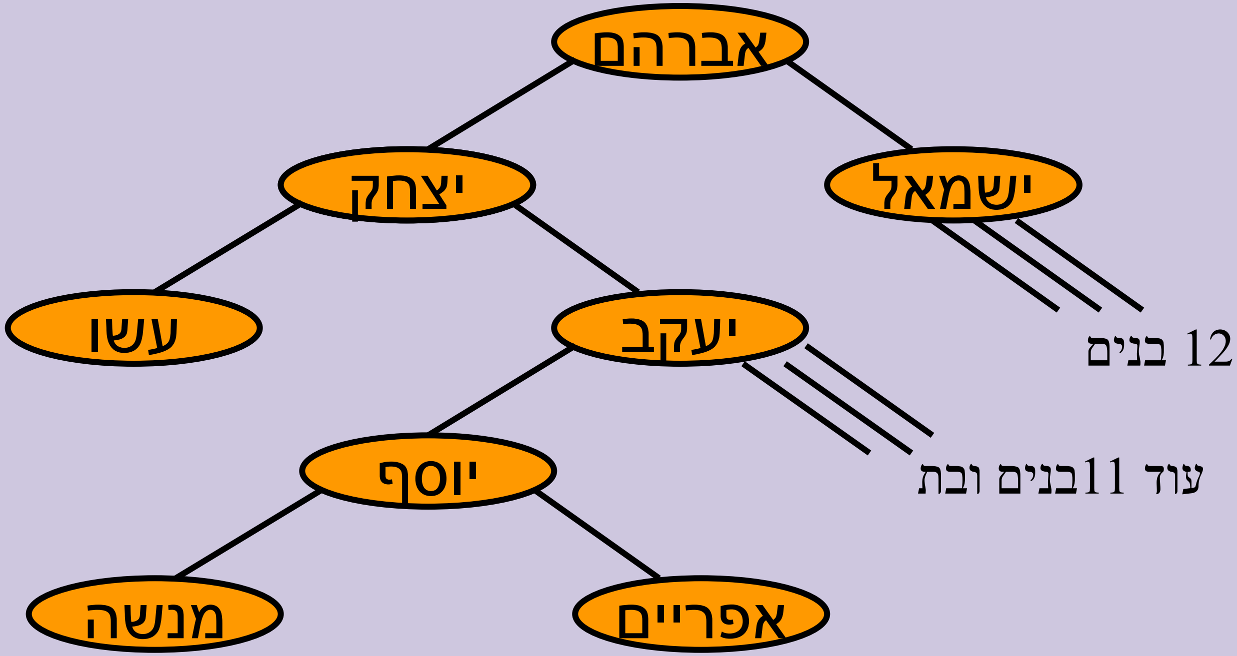
Chapter 5.5- Trees (91 - 97)

Chapter 13- Binary Search Trees (244 - 262)

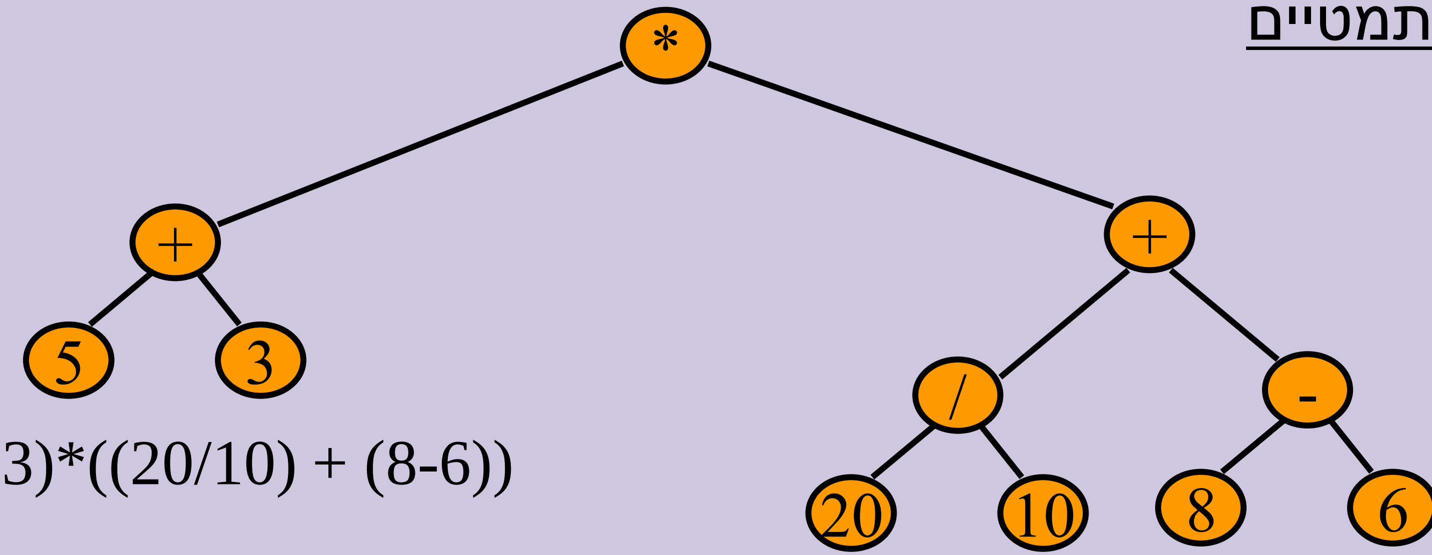
עצים

דוגמאות

1. אילן יוחסין



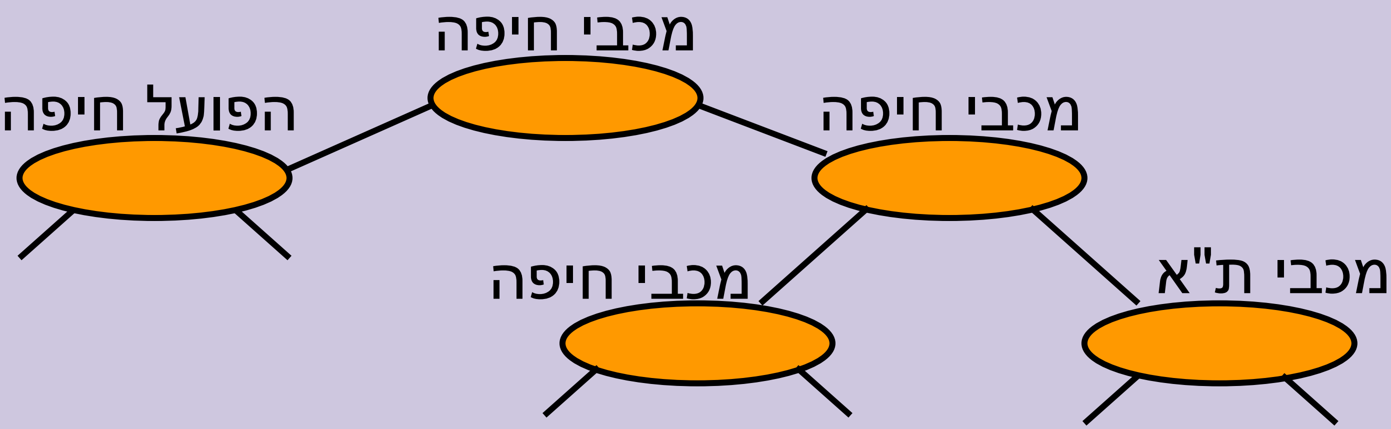
2. ביטויים אריתמטיים



(5+3)*((20/10) + (8-6))

עצים

3. עץ מנצחים (גביע)

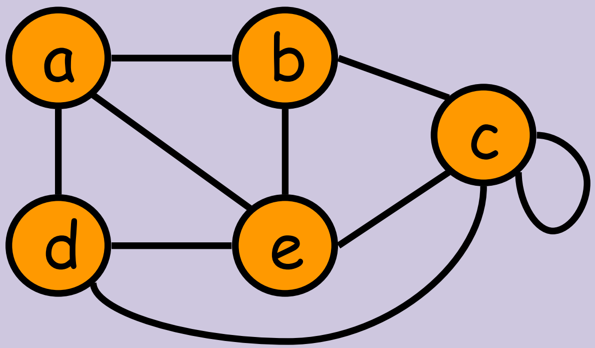


4. מבנה היררכי

צה"ל
חיל הים
חיל האוויר
חיל היבשה
שריון
תותחנים
...
חי"ר
חטיבת צנחנים
חטיבת הנח"ל
גדוד 856
...
גדוד 934
מחלקה 1
מחלקה 2
כיתה 1
כיתה 2
אלון אבוטבול

גרפים לא-מכוונים (Undirected Graphs)

גרף לא-מכוון הוא זוג (V,E) המורכב מקבוצת צמתים V וקבוצת קשתות E.
קשת ב-E היא קבוצה בת שני איברים מתוך V. קשת מסומנת ע"י (i,j) (במקום הסימון המדויק יותר $\{i,j\}$).



חוג עצמי

$V = \{a,b,c,d,e\}$

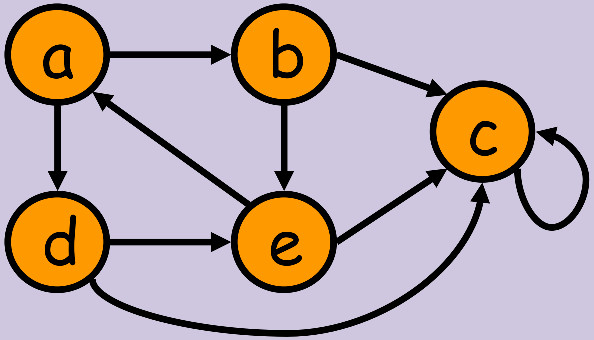
$E = \{(a,b), (a,d), (a,d) ,(b,c),(b,e),(c,c),(d,c),(d,e),(e,c) \}$

נסמן $|V| = n$ וכן $|E| = m$. בדוגמא: $n = 5$, $m = 9$.

מספר הקשתות m קטן בכל גרף מ- n^2 .

גרפים מכוונים (Directed Graphs)

גרף מכוון הוא זוג (V,E) המורכב מקבוצת צמתים V וקבוצת קשתות $E \subseteq V \times V$.



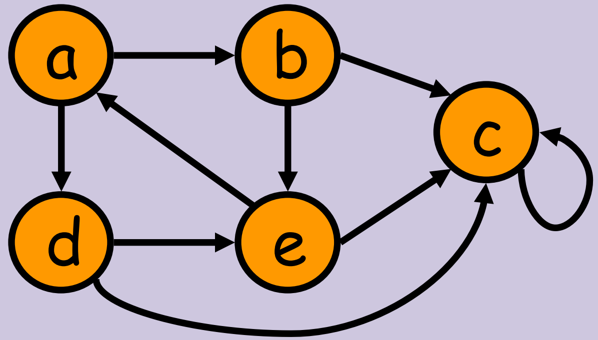
$V = \{a,b,c,d,e\}$

$E = \{(a,b),(a,d),(b,c),(b,e),(c,c),(d,c),(d,e),(e,a),(e,c)\}$

נסמן $n = |V|$ וכן $m = |E|$. בדוגמא: $n = 5$, $m = 9$.

גרפים מכוונים (Directed Graphs)

גרף מכוון הוא זוג (V,E) המורכב מקבוצת צמתים V וקבוצת קשתות $E \subseteq V \times V$.



$V = \{a,b,c,d,e\}$

$E = \{(a,b),(a,d),(b,c),(b,e),(c,c),(d,c),(d,e),(e,a),(e,c)\}$

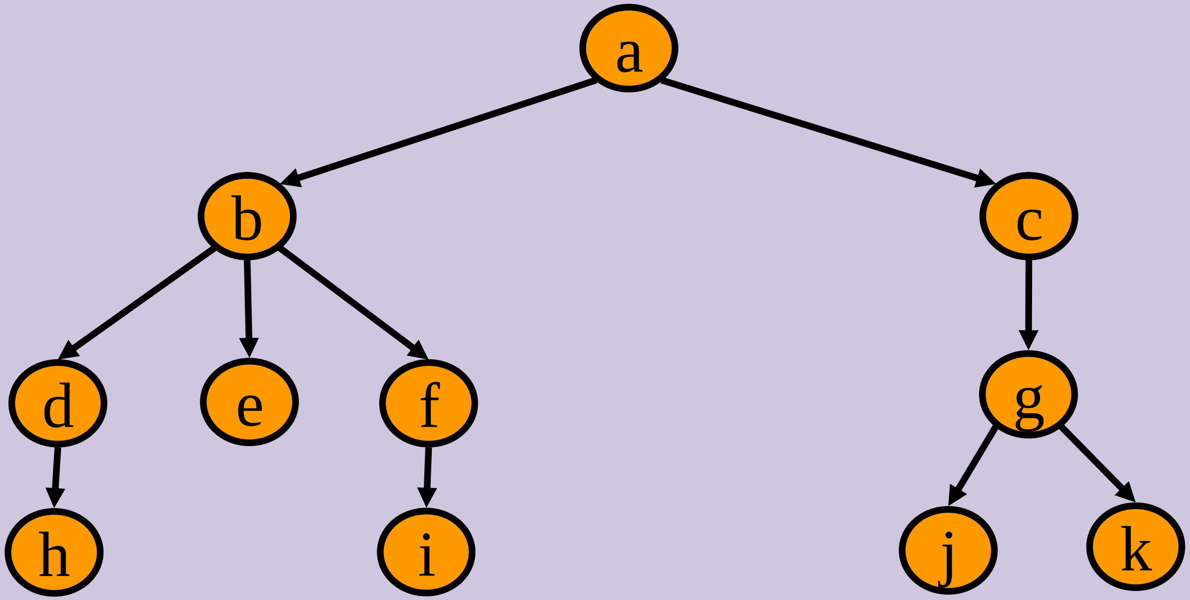
נסמן $|V| = n$ וכן $|E| = m$. בדוגמא: $n = 5$, $m = 9$.

<u>מסלול</u> (מכוון) בגרף (מכוון) (V,E) הוא סדרת צמתים $(v_1,v_2,..,v_k)$ כך שלכל זוג צמתים עוקבים בסדרה, (v_i,v_{i+1}) היא קשת ב-E. המסלול נקרא **<u>מעגל</u>** (מכוון) אם $v_1=v_k$ (לדוגמא, (a,d,e,a)). **<u>גרף התשתית</u>** של גרף מכוון G הוא גרף לא-מכוון עם אותם צמתים כמו ב- G ואותם קשתות כמו ב-G אך ללא כוון. לדוגמא, הגרף בשקף הקודם הוא גרף התשתית של הגרף הנתון בשקף זה.

עצים מכוונים

<u>מקור</u> הוא צומת שאף קשת אינה מצביעה אליו.

<u>עץ מכוון</u> הוא גרף מכוון ללא מעגלים (בגרף התשתית שלו) ואשר לו מקור אחד בלבד הנקרא **<u>שורש</u>**.

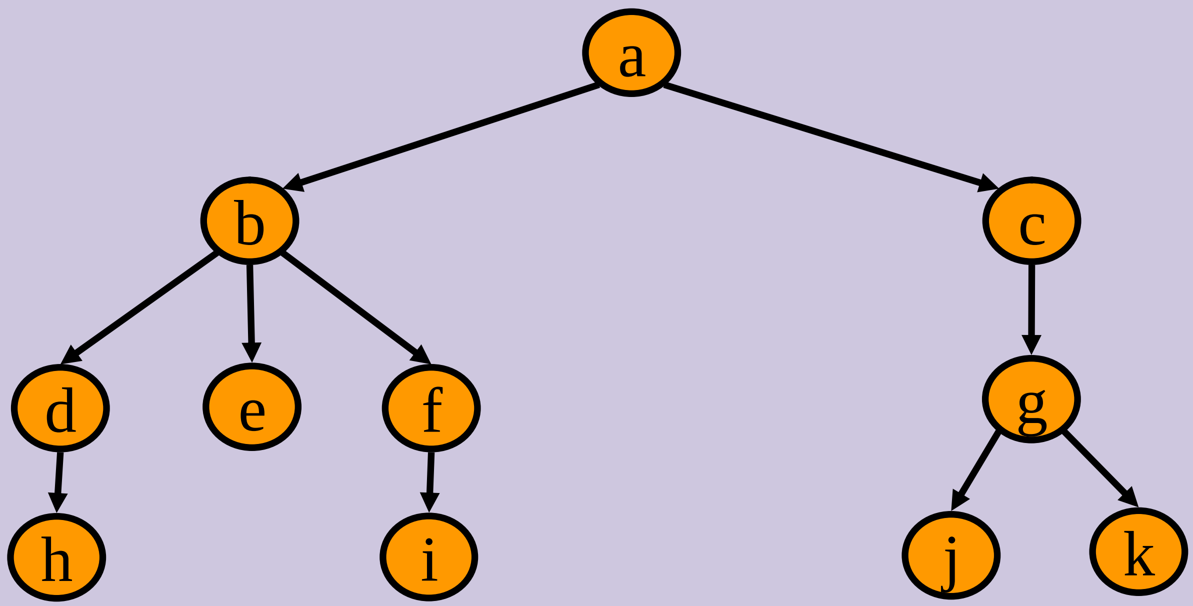


עצים מכוונים

מקור הוא צומת שאף קשת אינה מצביעה אליו.

עץ מכוון הוא גרף מכוון ללא מעגלים (בגרף התשתית שלו) ואשר לו מקור אחד בלבד הנקרא **שורש**.

הגדרות	דוגמאות
v **בן** של u אם קיימת קשת מצומת u לצומת v.	f בן של b
u **אב** של v אם v בן של u.	b אב של e



עצים מכוונים

<u>הגדרות</u>

v **<u>צאצא</u>** של u אם קיים מסלול מכוון מצומת u ל- v.

u **<u>אב-קדמון</u>** של v אם v צאצא של u.

<u>תת-עץ של G ששורשו v</u> הוא עץ מכוון שצמתיו הם v עצמו וכל הצאצאים של v, והקשתות שלו הן הקשתות המחברות צמתים אלו ב- G.

<u>דרגת צומת v</u> היא מספר הבנים של v.

<u>עלה</u> הוא צמת ללא בנים.

<u>צומת פנימי</u> הוא צומת שאינו עלה.

<u>דוגמאות</u>

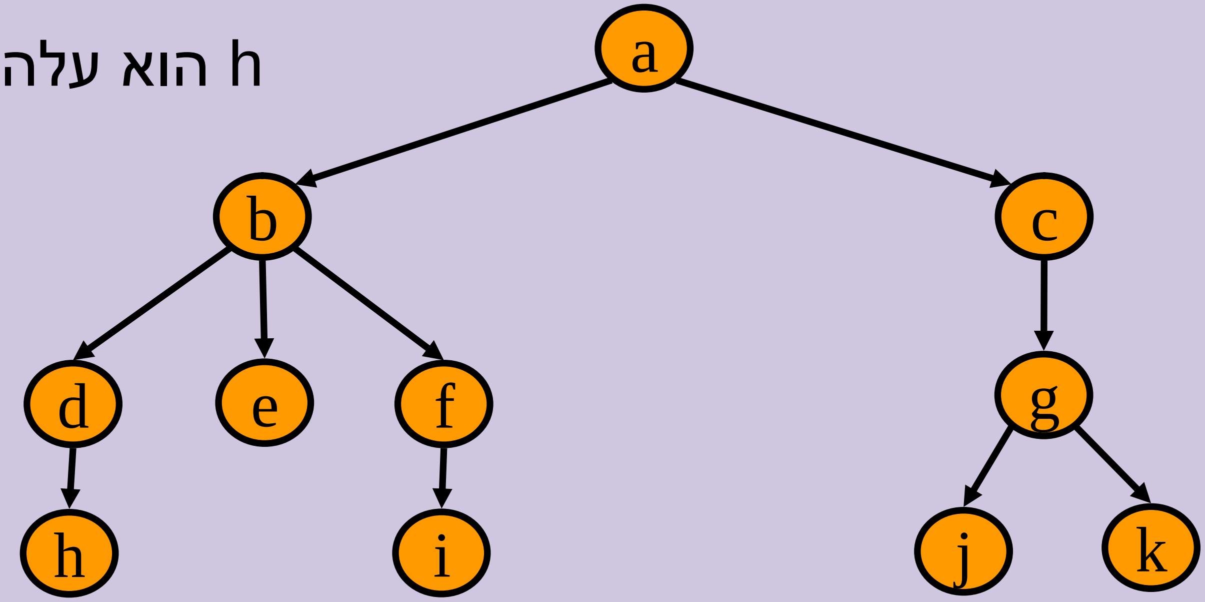
g צאצא של a

b אב-קדמון של h

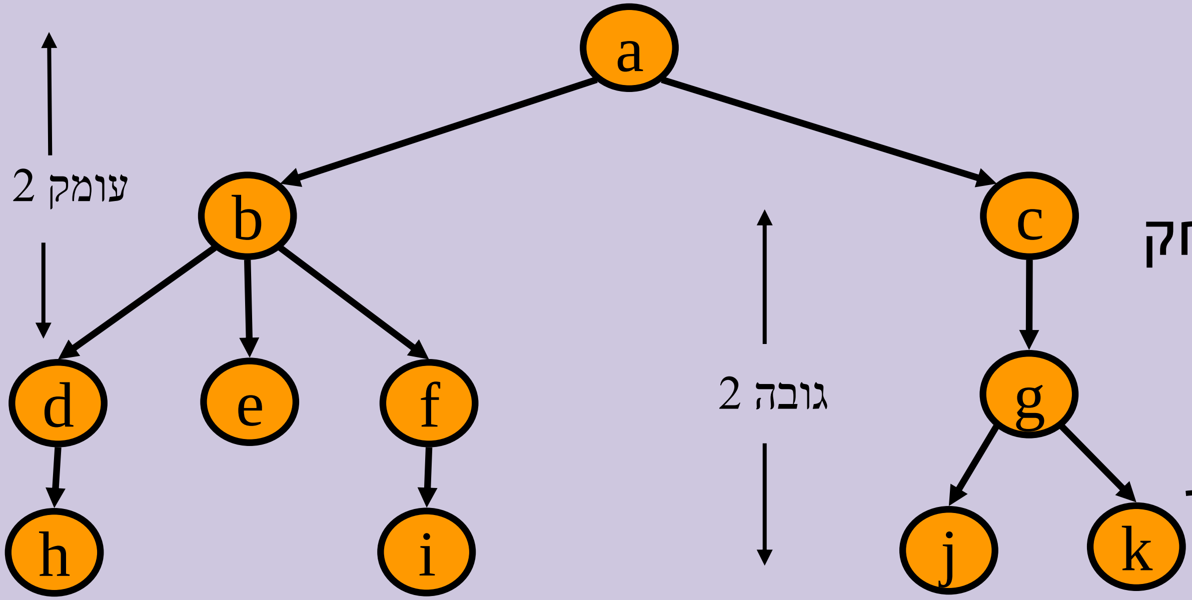
תת העץ ששורשו g מכיל 3 צמתים ושתי קשתות.

דרגת a היא 2.

h הוא עלה.



עצים מכוונים

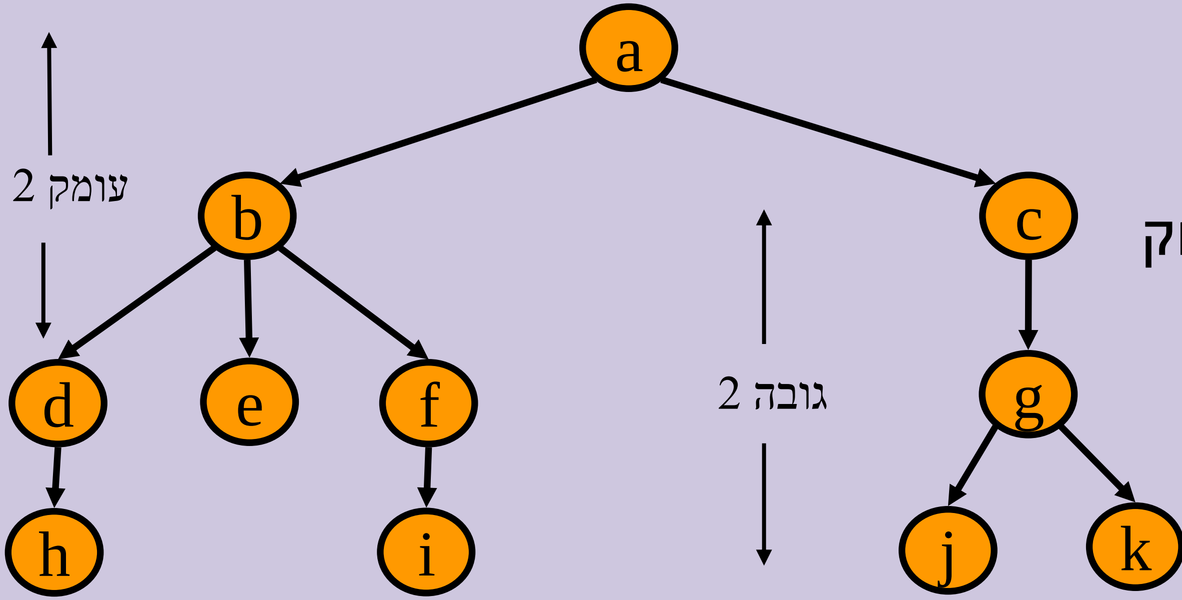


<u>עומק של צומת v</u> הוא מספר הקשתות משורש העץ אל v (המרחק מהשורש).

<u>גובה של צומת v</u> הוא מספר הקשתות מ-v לצאצא הרחוק ביותר של v (עלה).

<u>גובה העץ</u> הוא הגובה של שורשו.

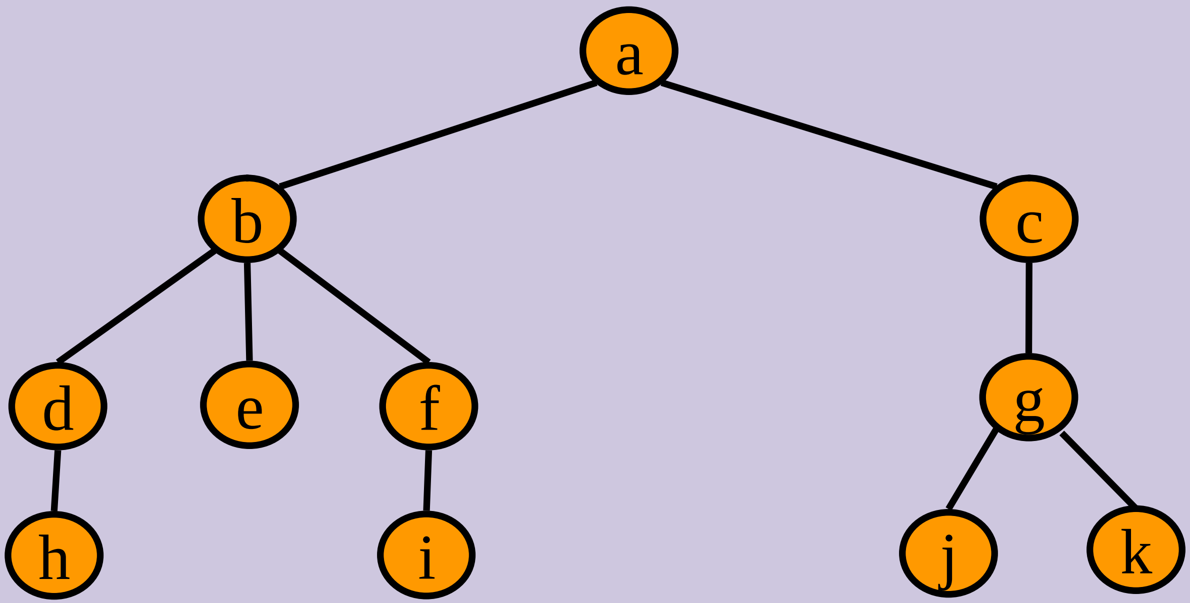
עצים מכוונים



<u>עומק של צומת v</u> הוא מספר הקשתות משורש העץ אל v (המרחק מהשורש).

<u>גובה של צומת v</u> הוא מספר הקשתות מ-v לצאצא הרחוק ביותר של v (עלה).

<u>גובה העץ</u> הוא הגובה של שורשו.

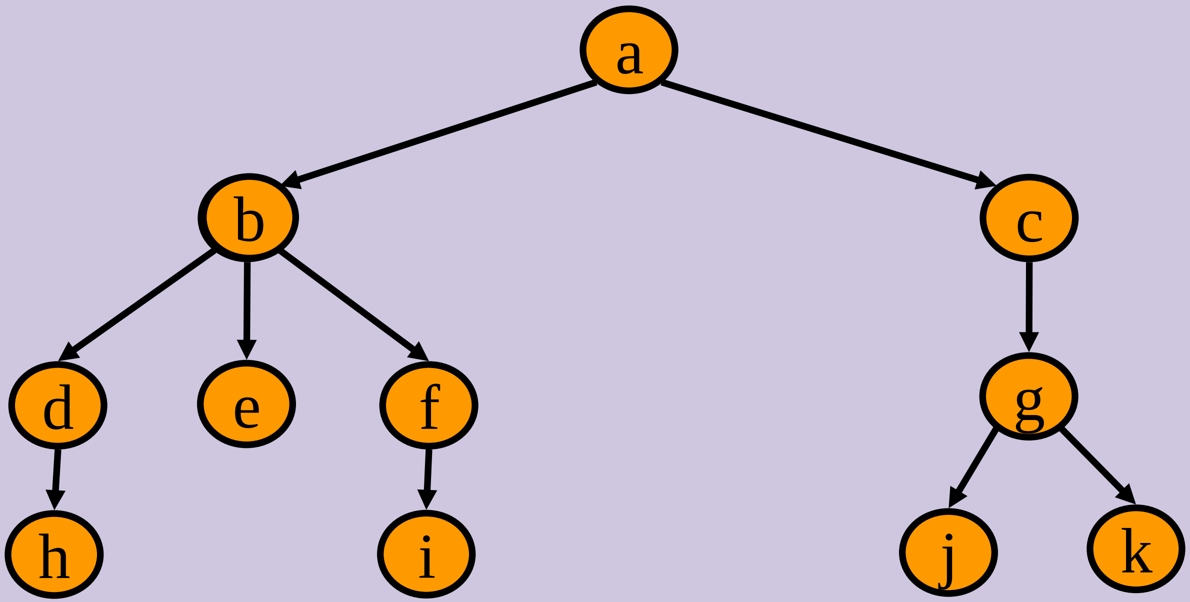


<u>הערה:</u> לעיתים נשמיט את החצים מתוך הבנה שכוון הקשתות כלפי מטה. כמו כן לרוב נאמר עץ במקום עץ מכוון.

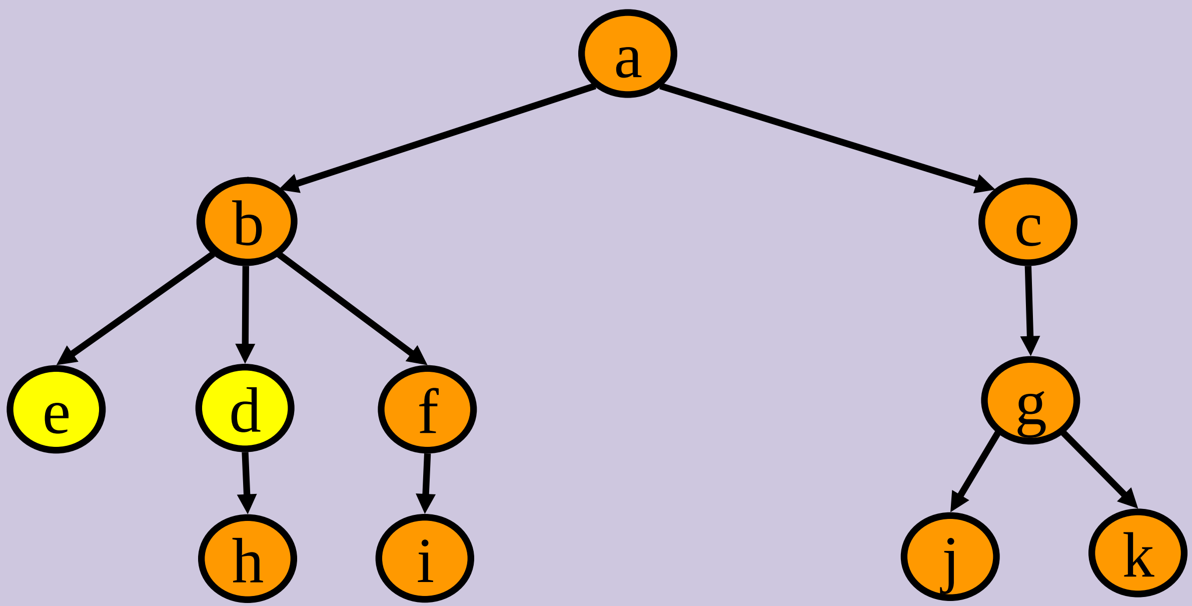
עצים מסודרים

עץ מסודר הוא עץ מכוון

שבו הבנים של כל צומת מסודרים (משמאל לימין).

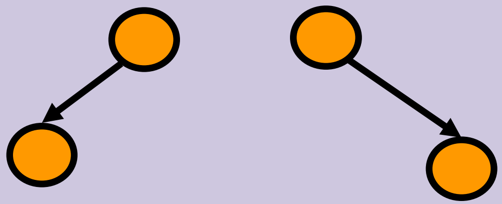


למשל עץ זה שונה מהעץ העליון בגלל שסדר הבנים השתנה.



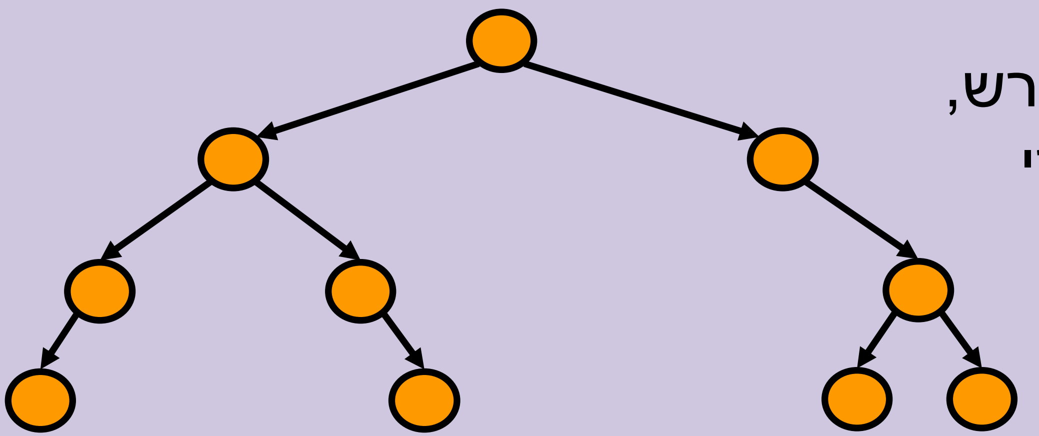
<u>עצים בינריים</u>

<u>עץ בינרי</u>: עץ שבו לכל צומת שאינו עלה יש בן שמאלי ו/או בן ימני.

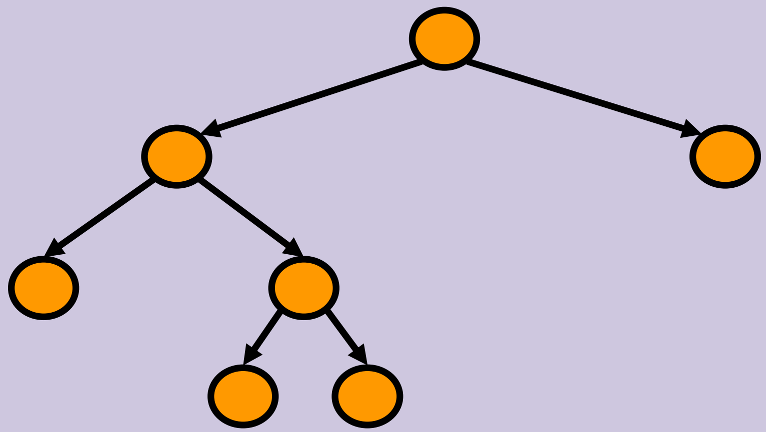


<u>הגדרה רקורסיבית</u>: <u>עץ בינרי</u> הוא מבנה

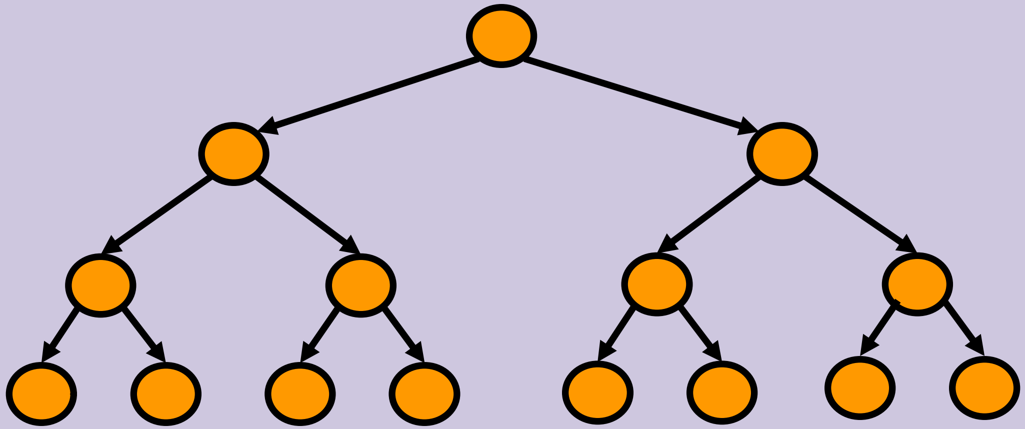
1. ריק (ללא צמתים), או
2. מורכב משלושה חלקים: צומת הנקרא שורש, עץ בינרי הנקרא תת-עץ שמאלי, ועץ בינרי הנקרא תת-עץ ימני.



עצים בינריים מלאים ושלמים



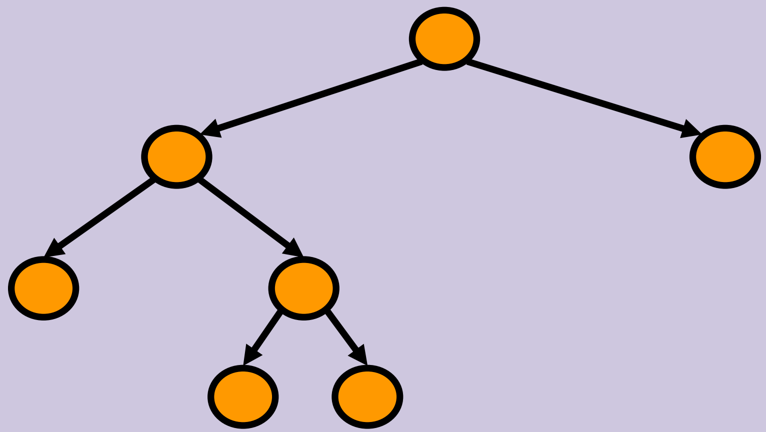
עץ בינרי מלא (full): עץ שבו לכל צומת פנימי 2 בנים.



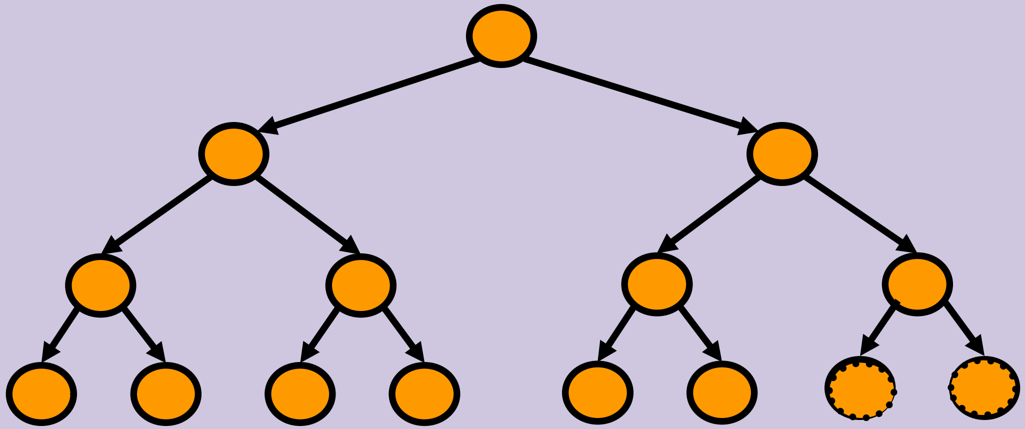
עץ בינרי שלם (complete): עץ בינרי מלא שבו כל העלים באותו עומק.

עצים בינריים מלאים ושלמים

עץ בינרי מלא (full): עץ שבו לכל צומת פנימי 2 בנים.



עץ בינרי שלם (complete): עץ בינרי מלא שבו כל העלים באותו עומק.

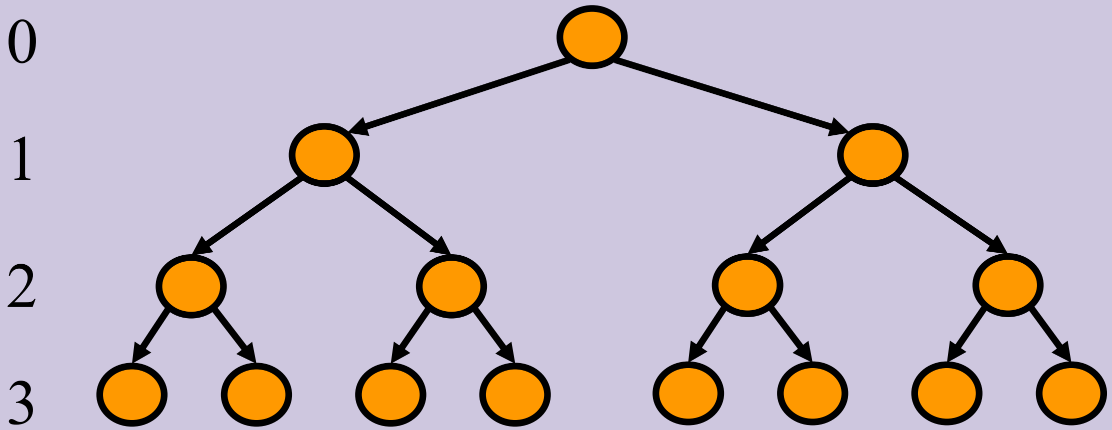


עץ בינרי כמעט שלם: עץ בינרי שלם שהוצאו ממנו עלים ("מצד ימין").

תכונות עצים בינריים שלמים

בעץ בינרי שלם בעל n צמתים, L עלים, וגובה h:

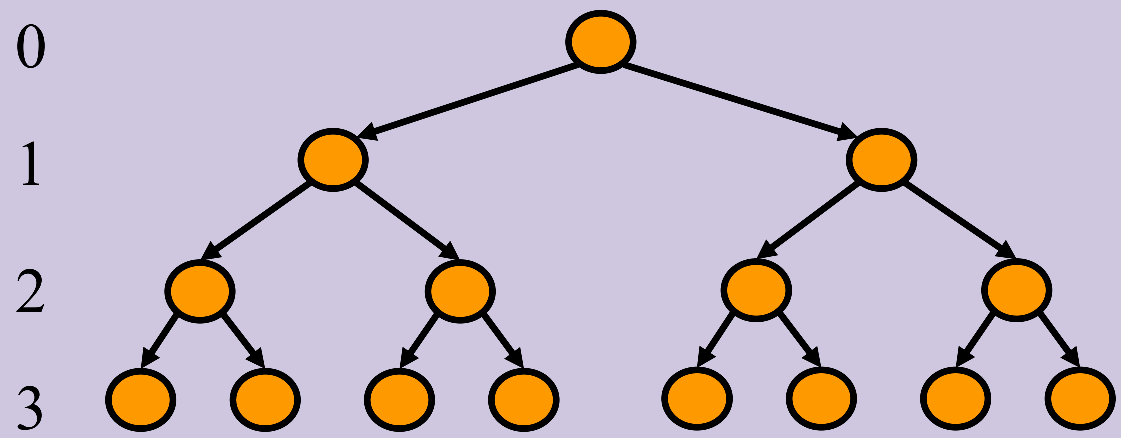
1. מספר הצמתים בעומק i: $n_i = 2^i$
2. מספר העלים: $L = n_h = 2^h$
3. מספר הצמתים: $n = \sum_{i=0}^{h} n_i = \sum_{i=0}^{h} 2^i = 2^{h+1} - 1$



תכונות עצים בינריים שלמים

בעץ בינרי שלם בעל n צמתים, L עלים, וגובה h:

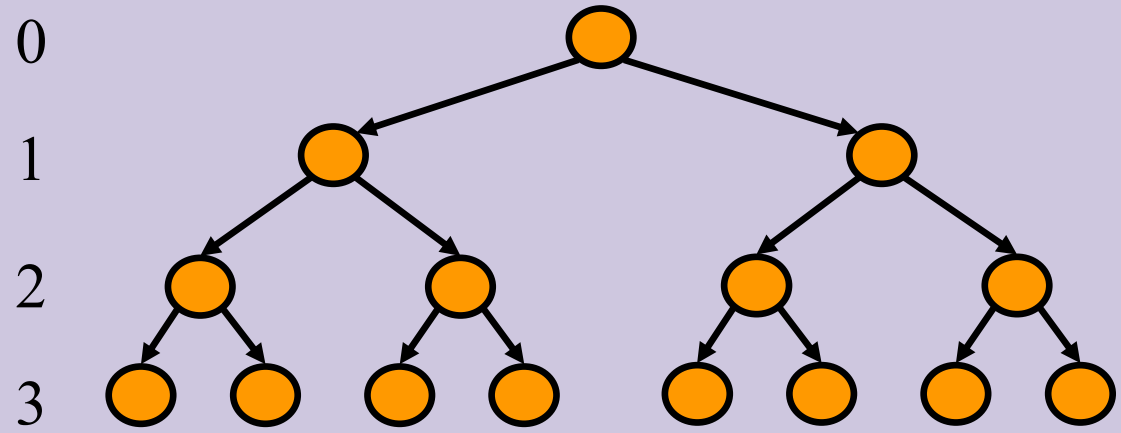
1. מספר הצמתים בעומק i: $n_i = 2^i$
2. מספר העלים: $L = n_h = 2^h$
3. מספר הצמתים: $n = \sum_{i=0}^{h} n_i = \sum_{i=0}^{h} 2^i = 2^{h+1} - 1$
4. הגובה: $h = \log_2(n+1) - 1$



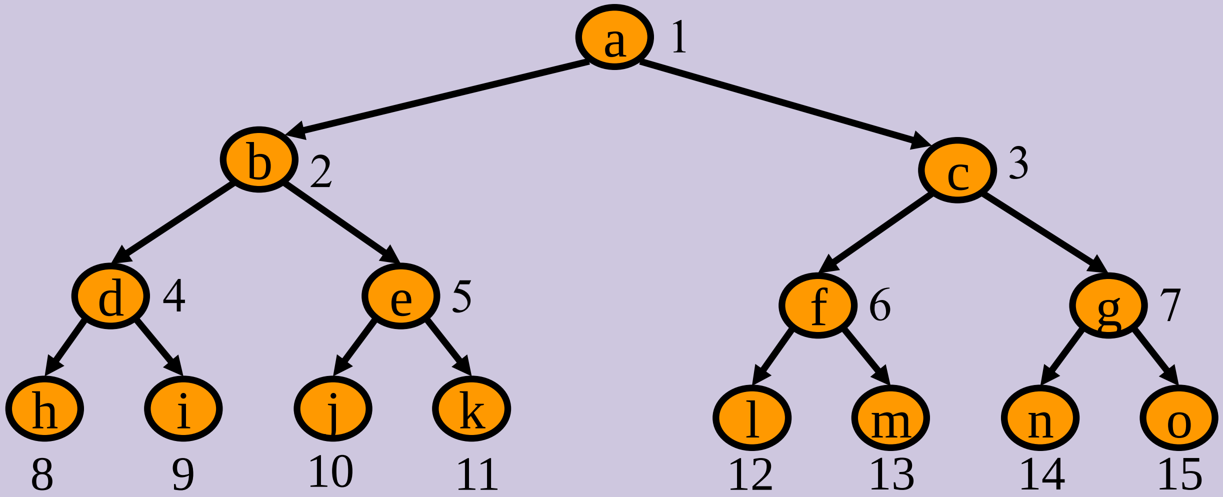
תכונות עצים בינריים שלמים

בעץ בינרי שלם בעל n צמתים, L עלים, וגובה h:

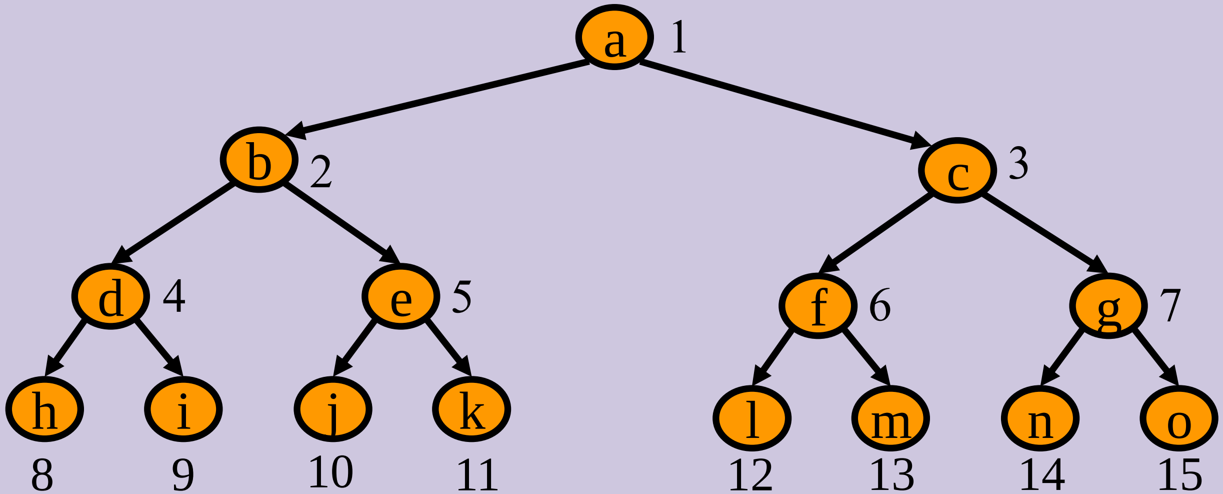
1. מספר הצמתים בעומק i: $n_i = 2^i$
2. מספר העלים: $L= n_h= 2^h$
3. מספר הצמתים: $n = \sum_{i=0}^{h} n_i = \sum_{i=0}^{h} 2^i = 2^{h+1} - 1$
4. הגובה: $h = \log_2(n+1) - 1$
5. מספר הצמתים הפנימיים: $n - L = 2^h - 1 = L - 1$



ייצוג לעצים בינריים שלמים



ייצוג לעצים בינריים שלמים



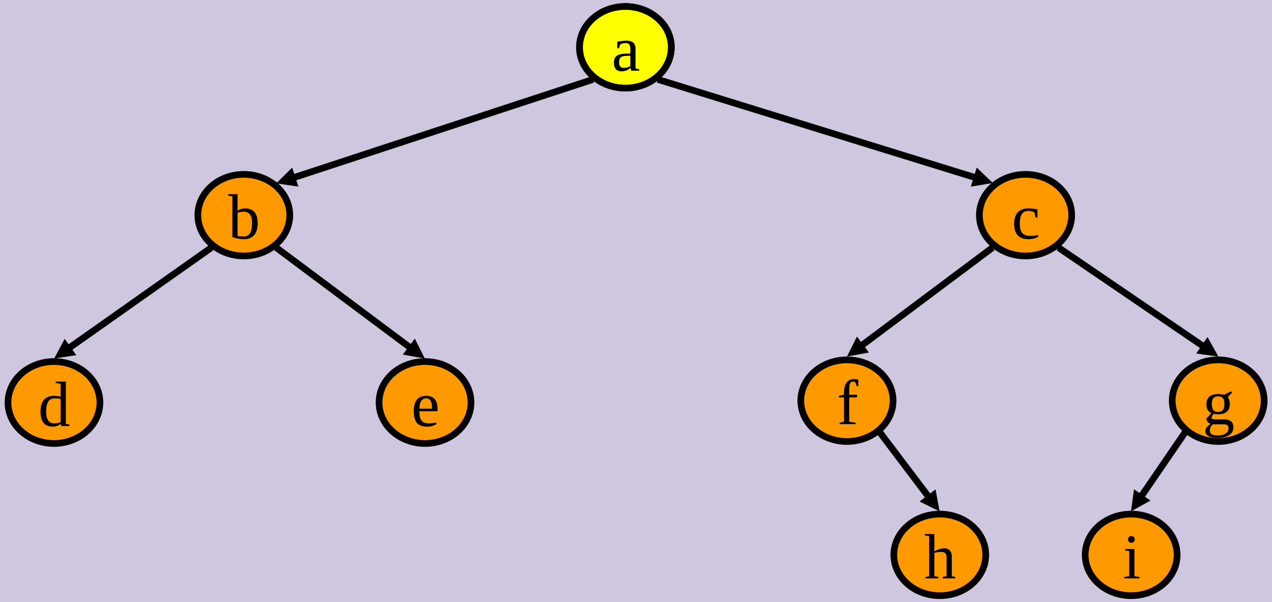
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o

בן שמאלי של צומת i
נמצא ב- 2i

בן ימני של צומת i נמצא
ב- 2i+1

אבא של צומת i
נמצא ב- $\lfloor i/2 \rfloor$

סיור בעצים בינריים



preorder

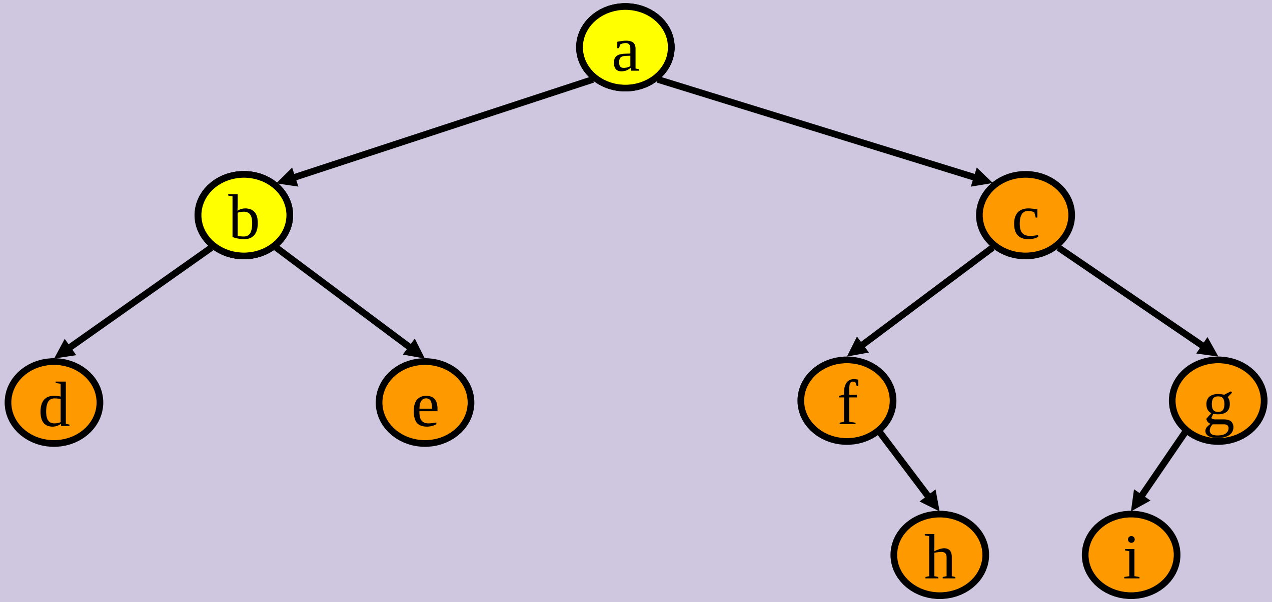
בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

סיור בעצים בינריים



preorder

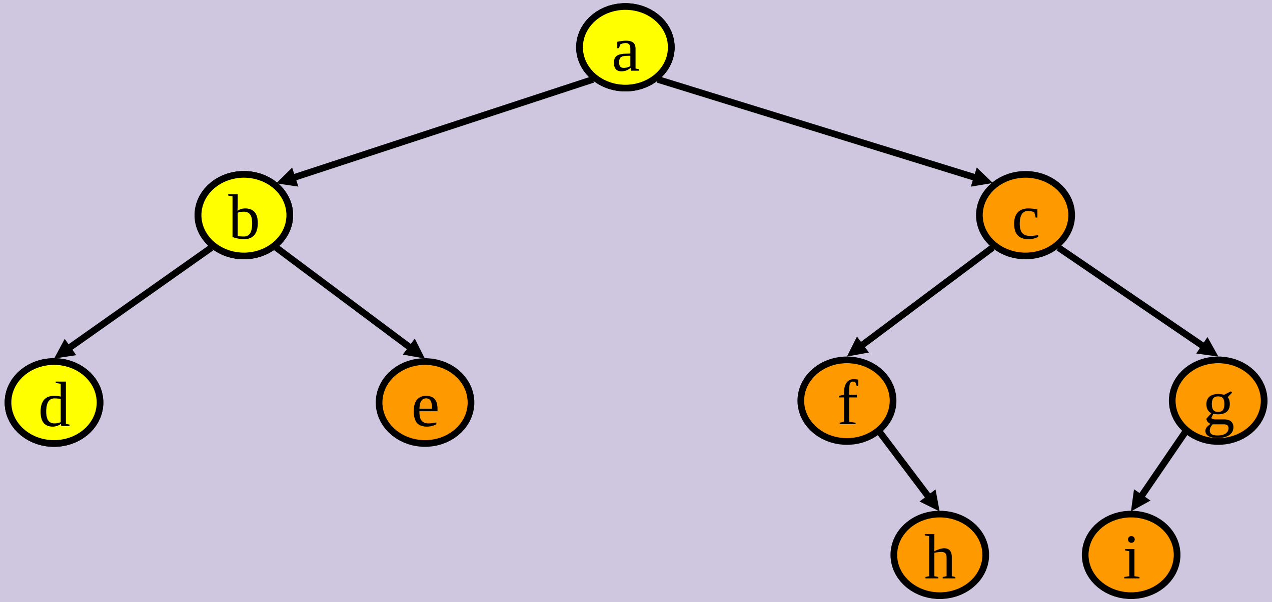
בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

סיור בעצים בינריים



preorder

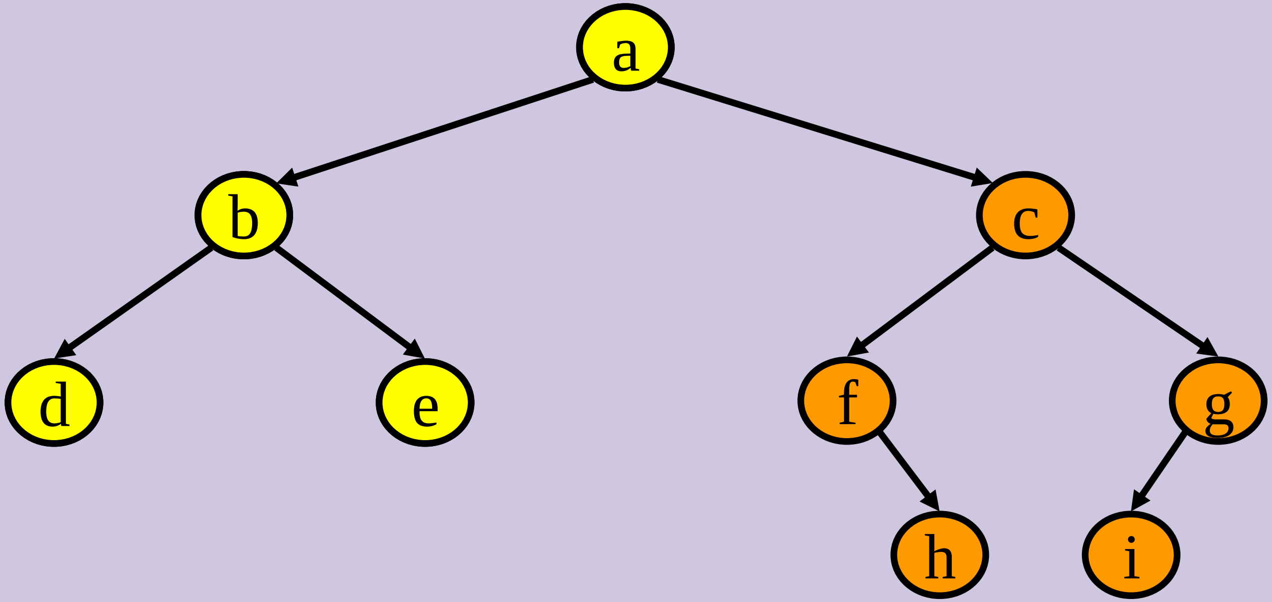
בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

סיור בעצים בינריים



preorder

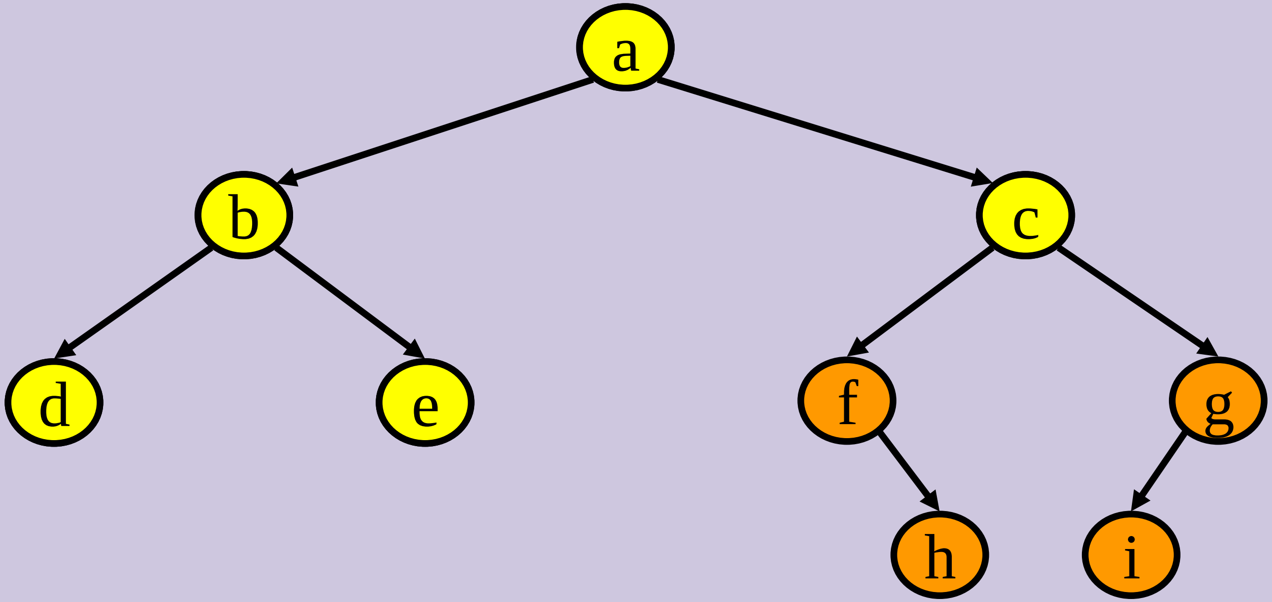
בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

סיור בעצים בינריים



preorder

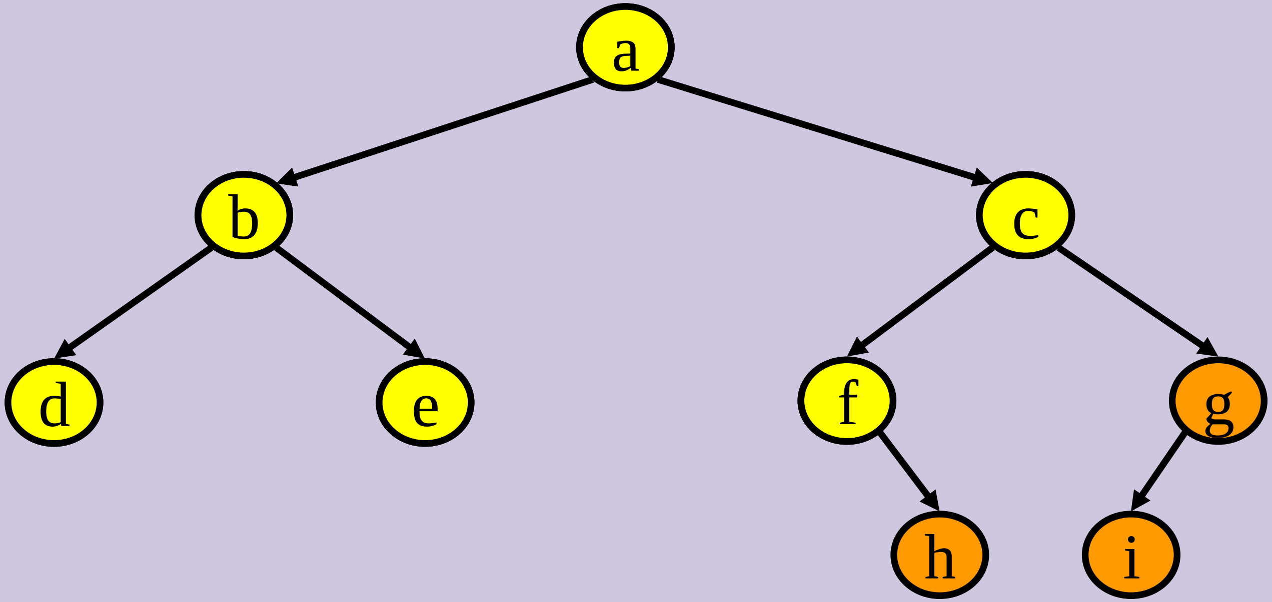
בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

סיור בעצים בינריים



preorder

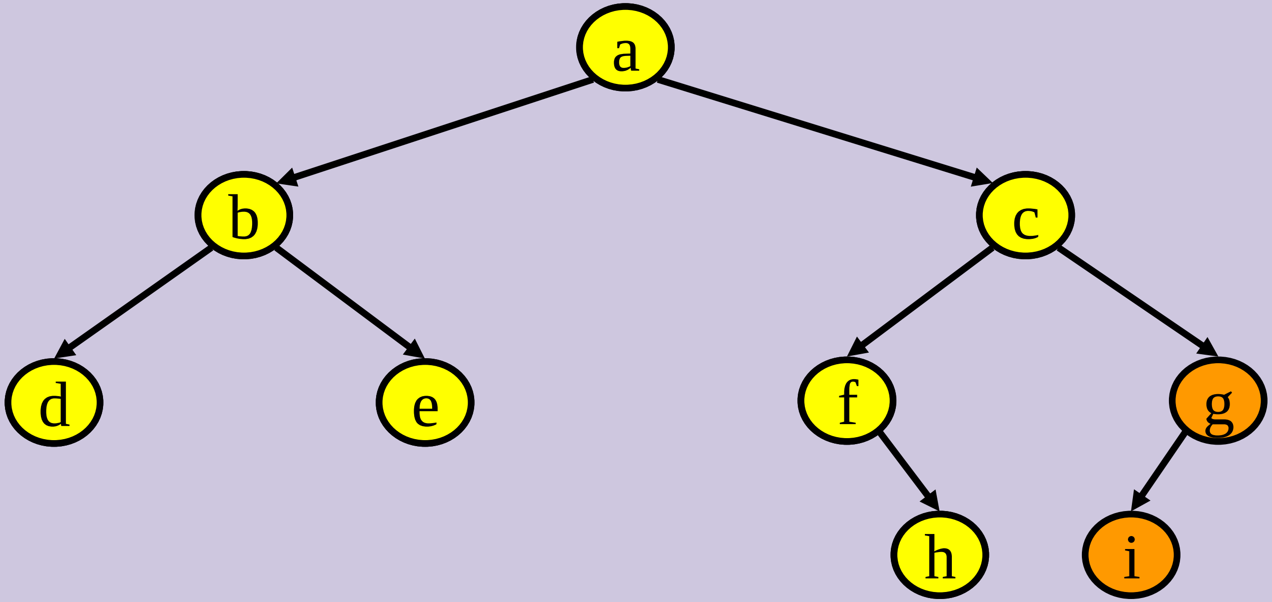
בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

סיור בעצים בינריים



preorder

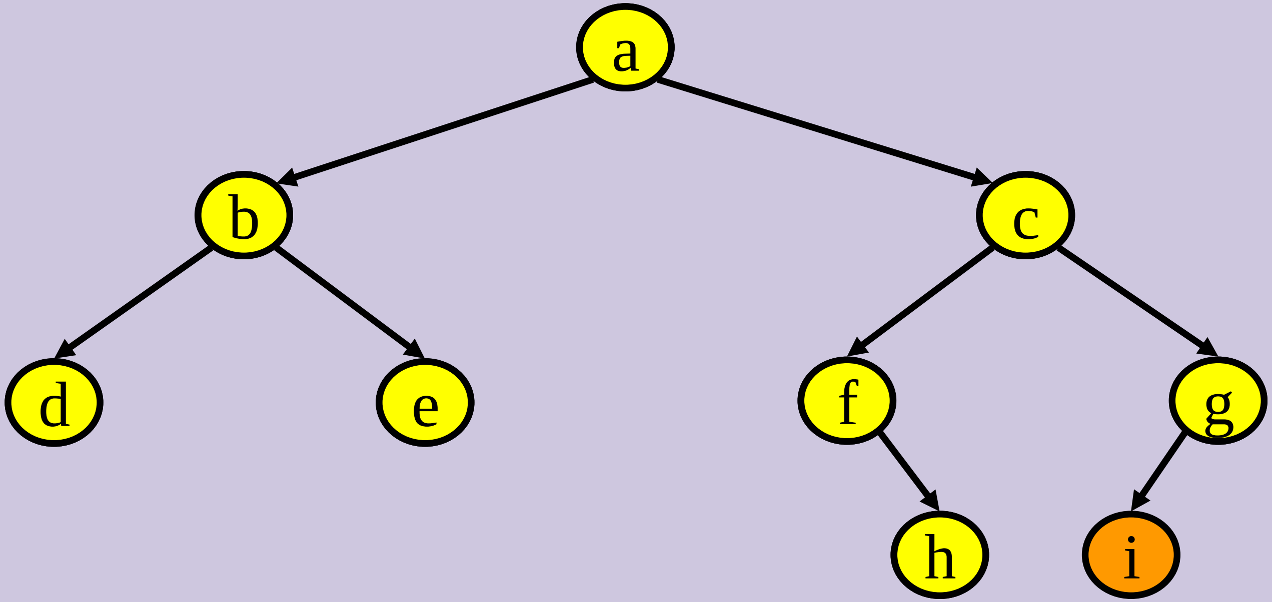
בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

סיור בעצים בינריים



preorder

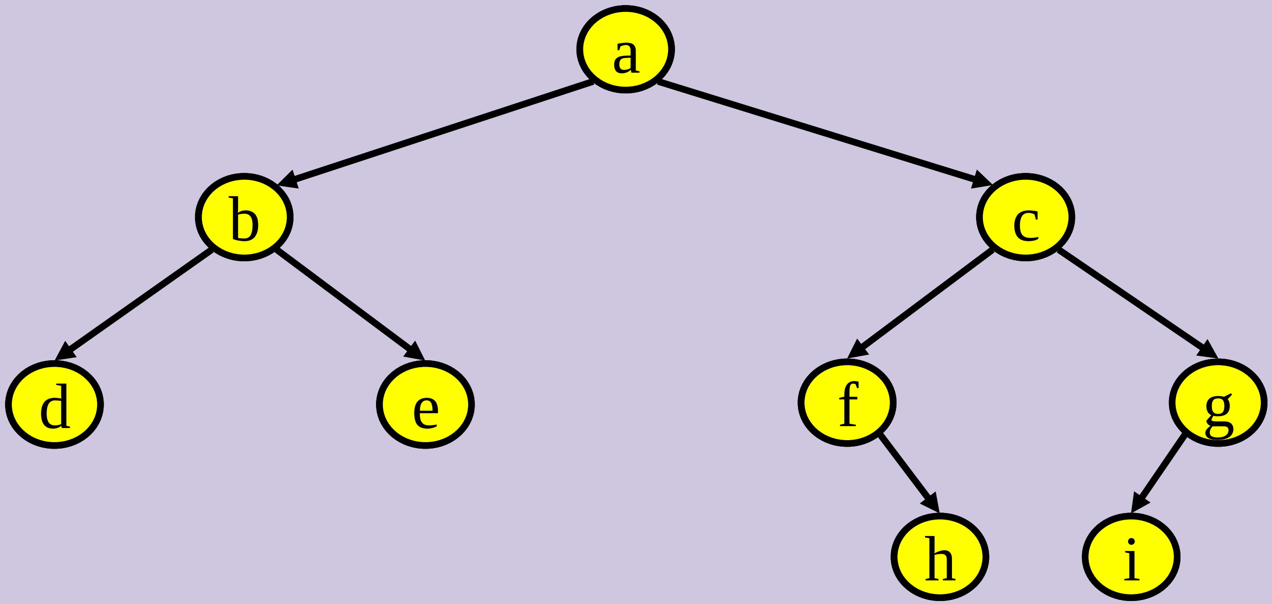
בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

סיור בעצים בינריים



preorder

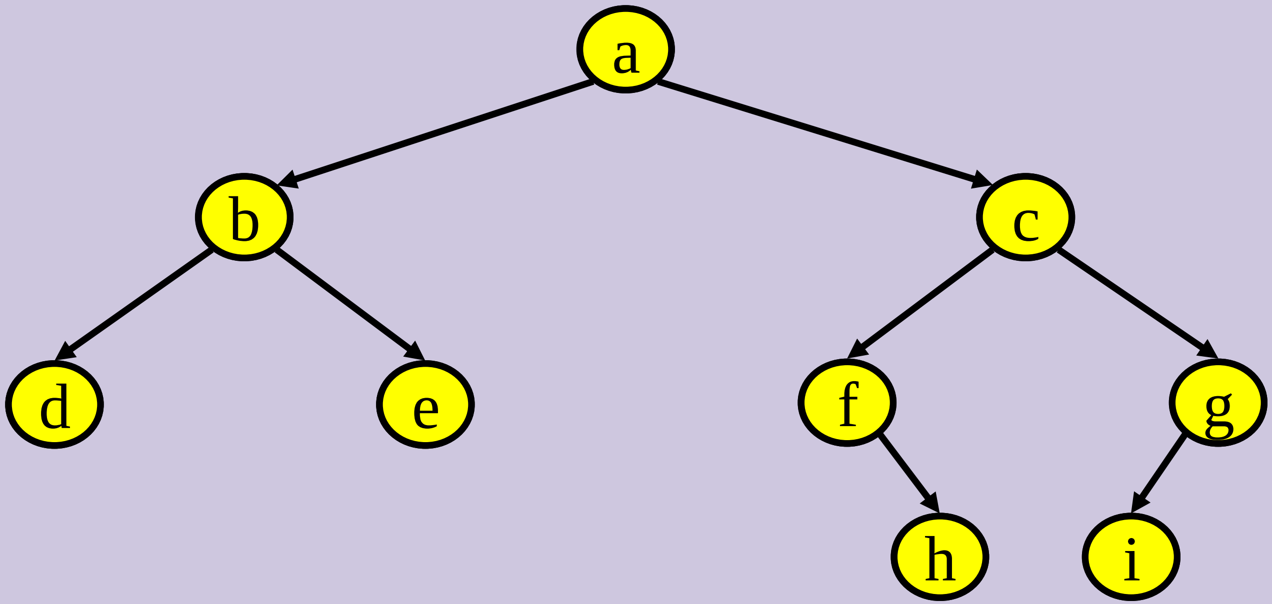
בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

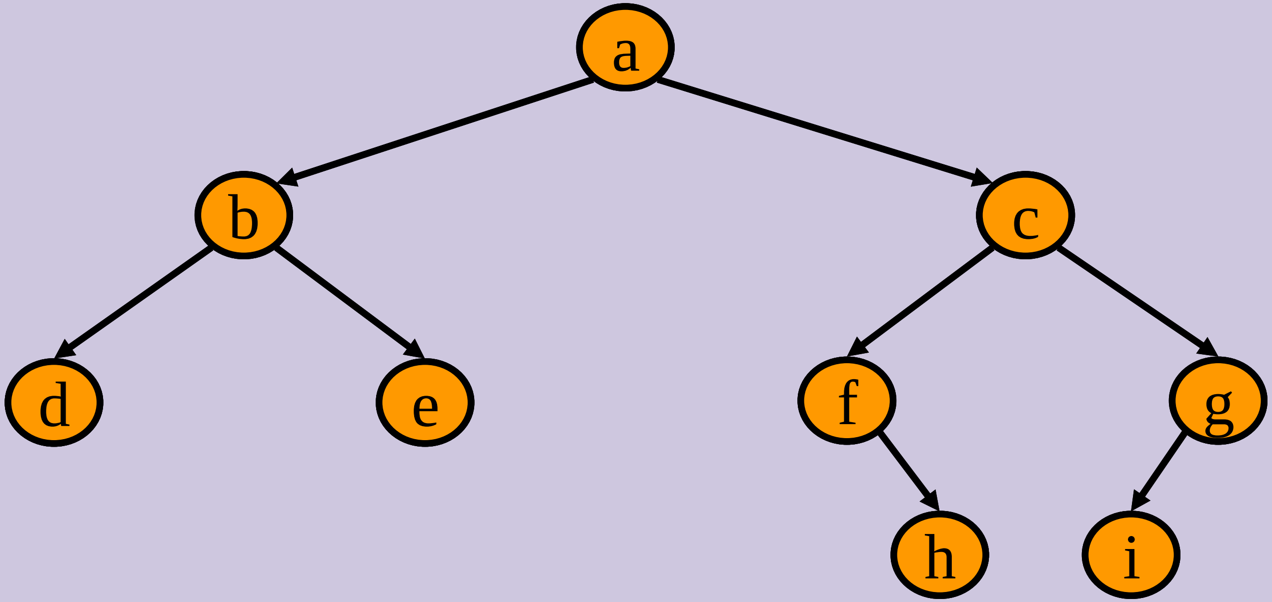
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

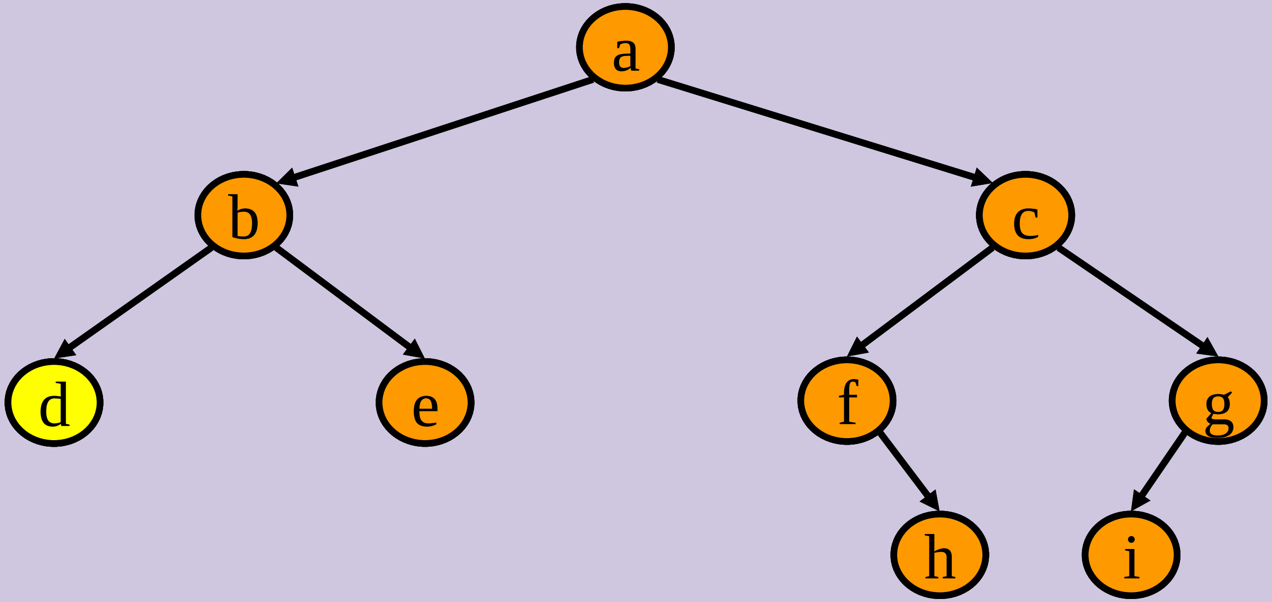
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

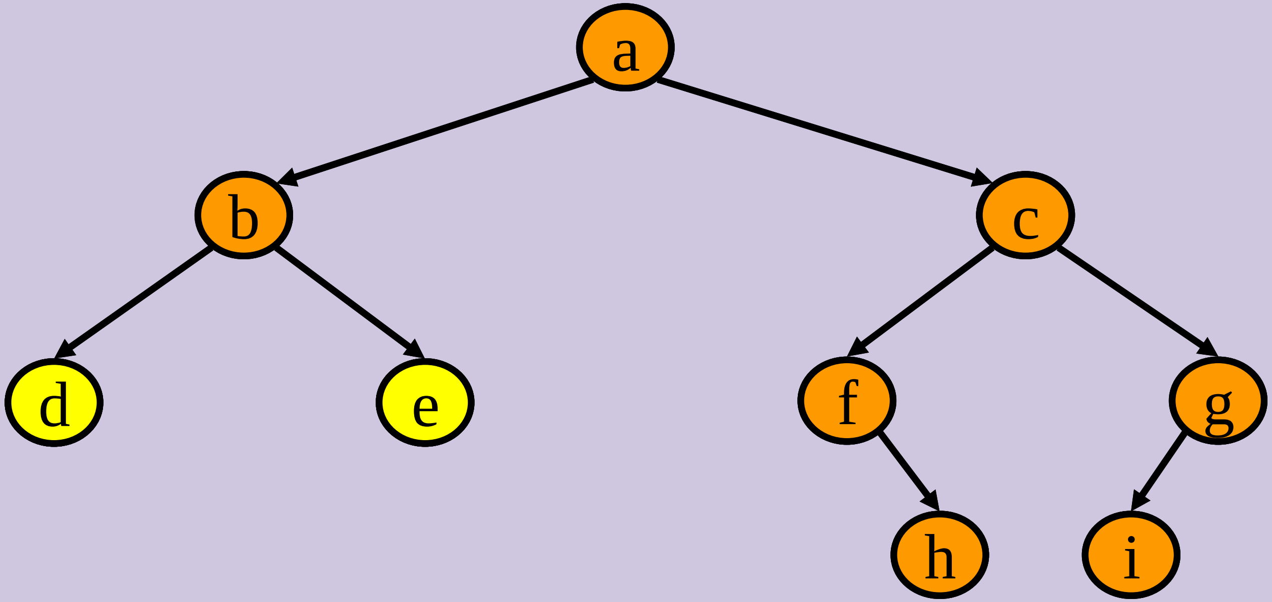
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

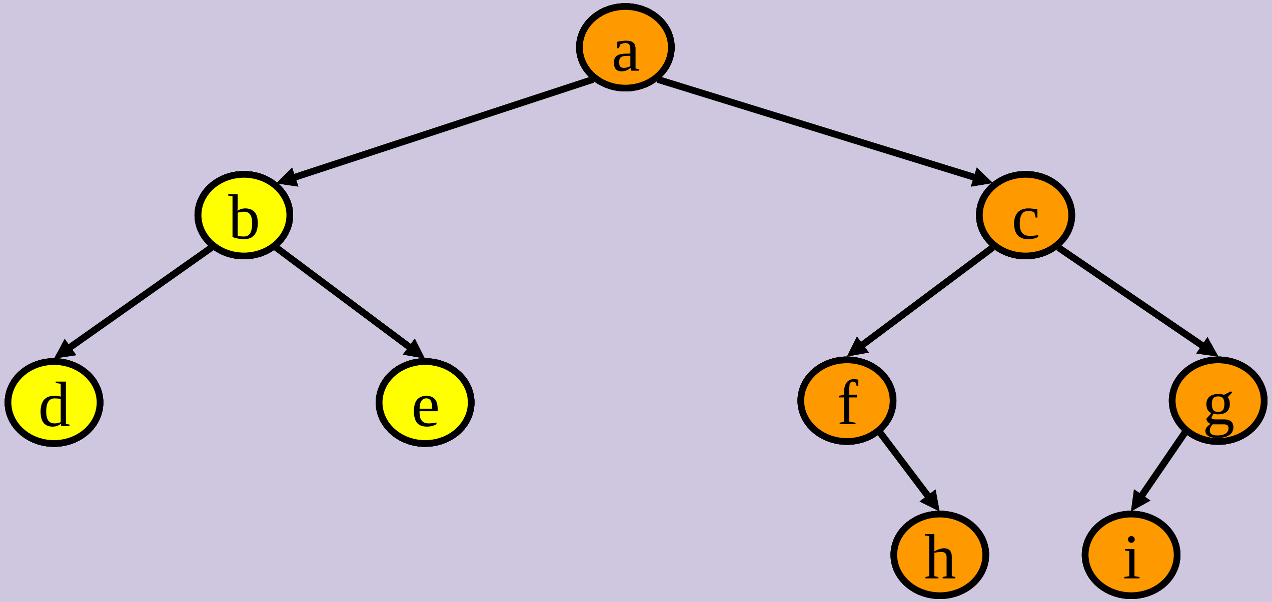
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

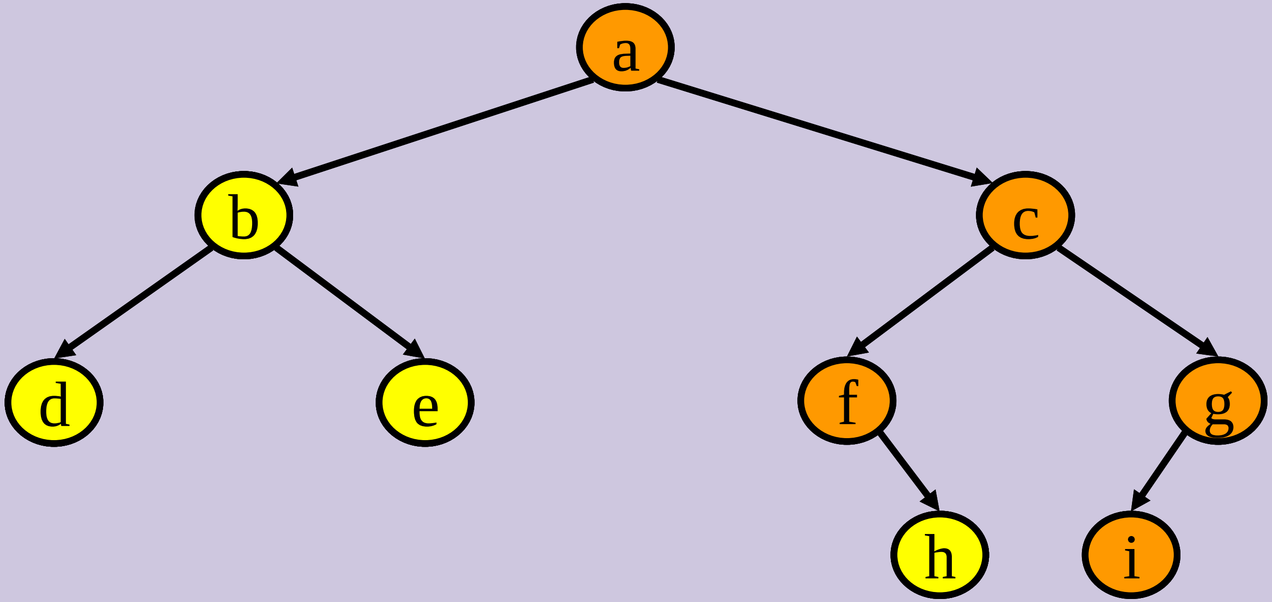
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

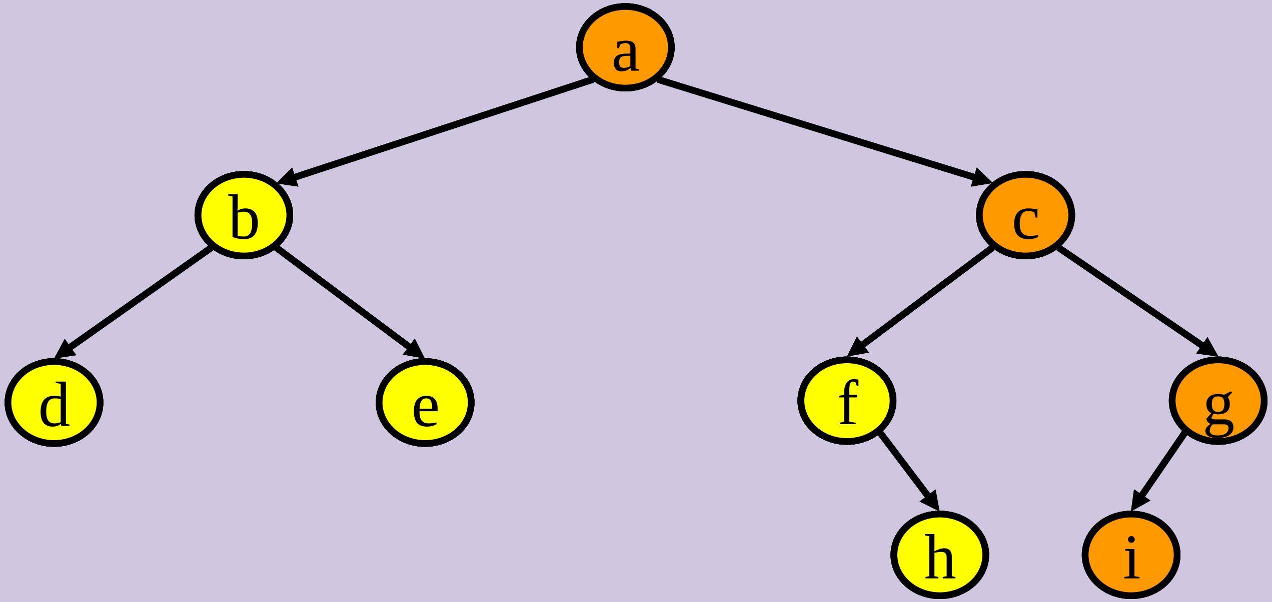
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

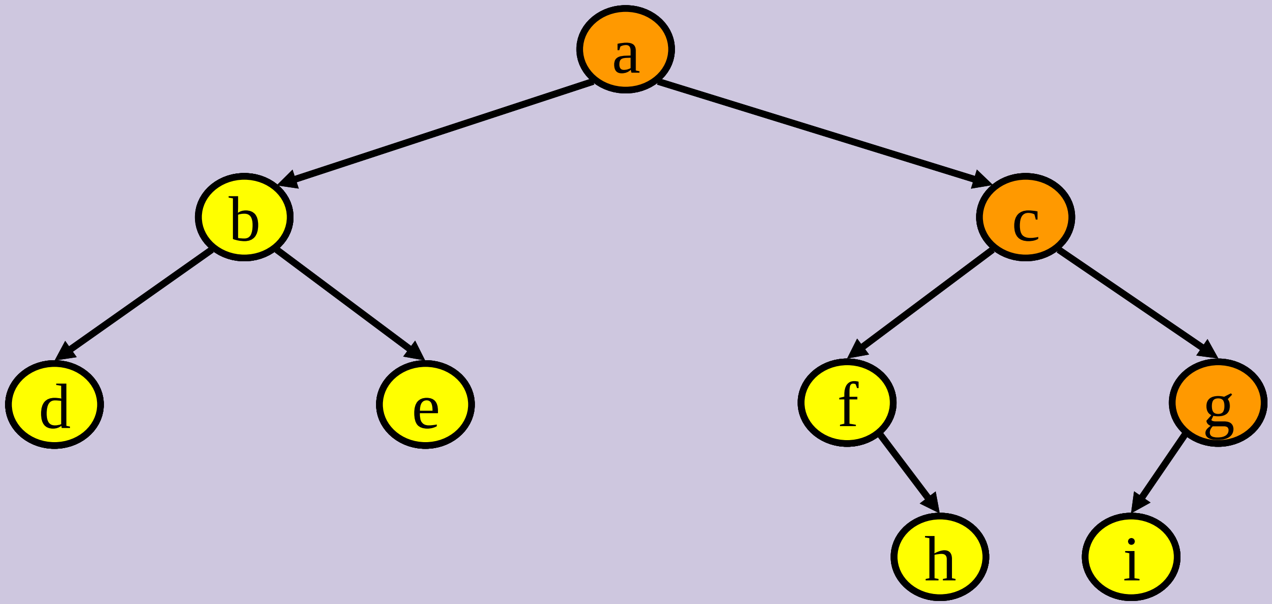
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

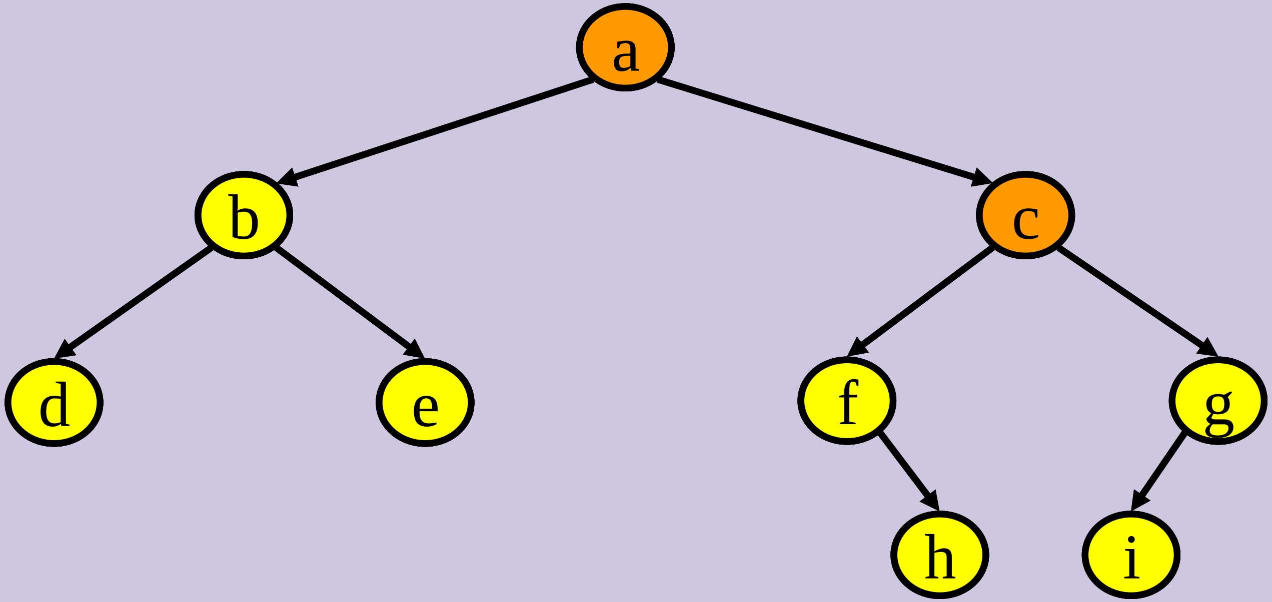
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

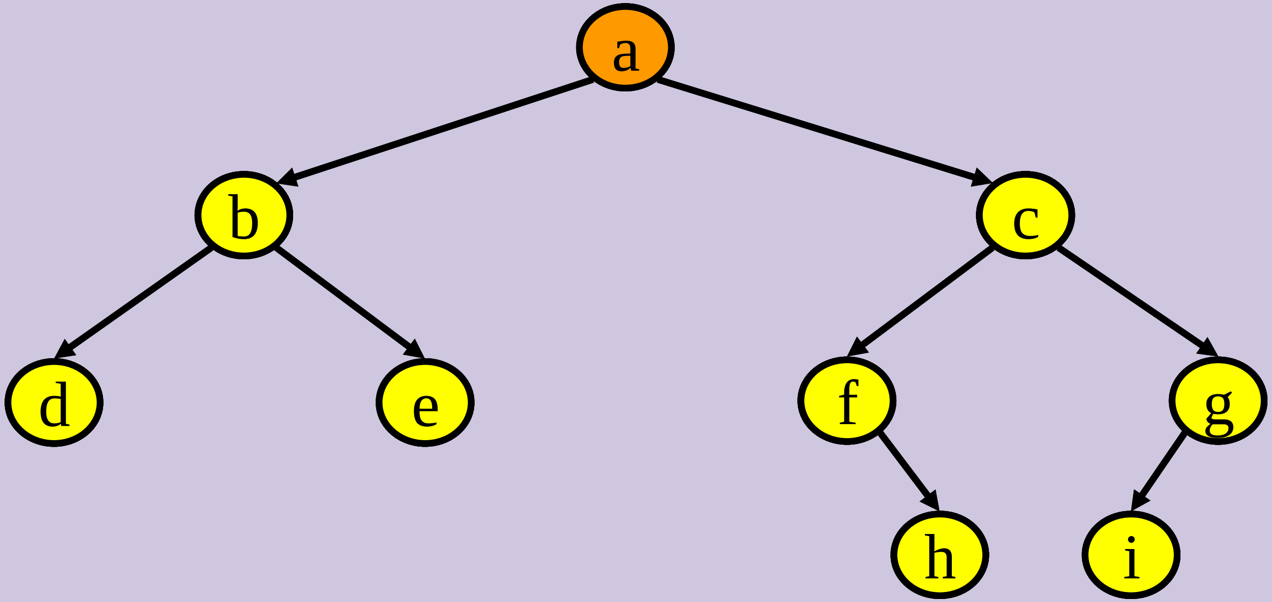
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

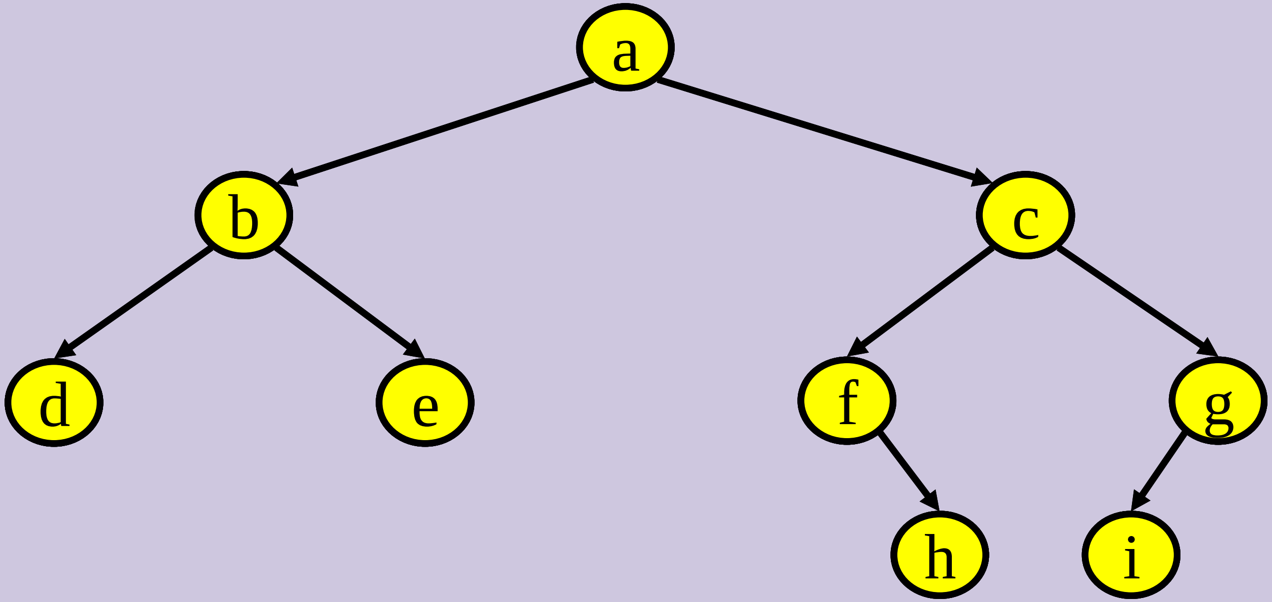
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

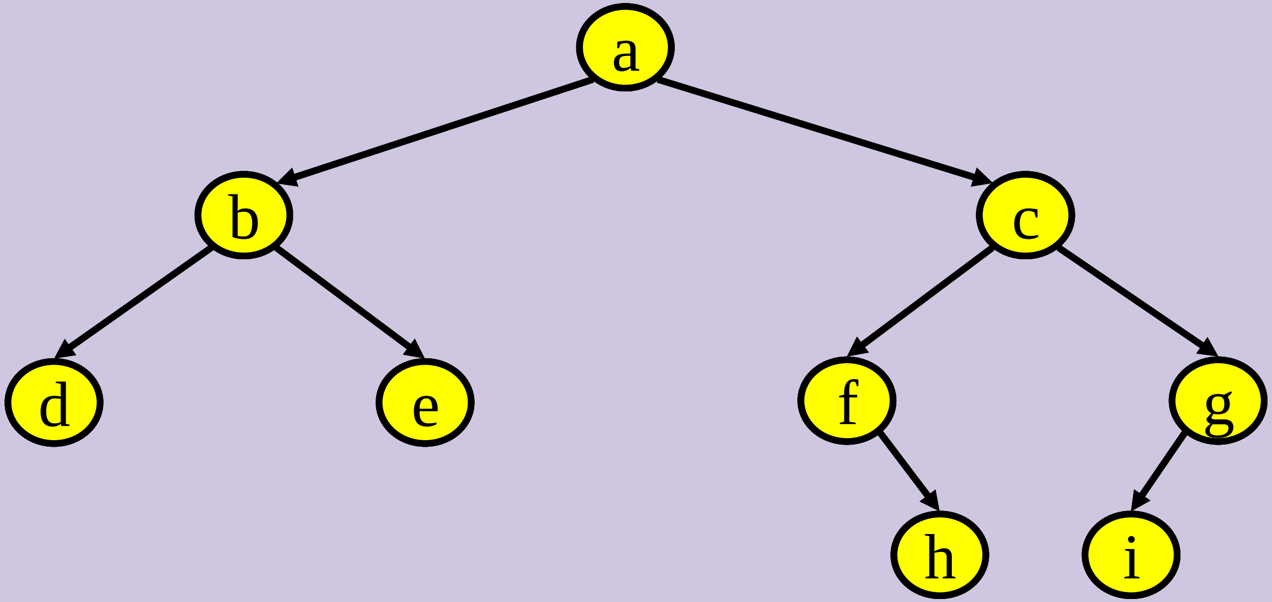
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

inorder

סייר בתת העץ השמאלי

בקר בשורש

סייר בתת העץ הימני

d b e a f h c i g

postorder

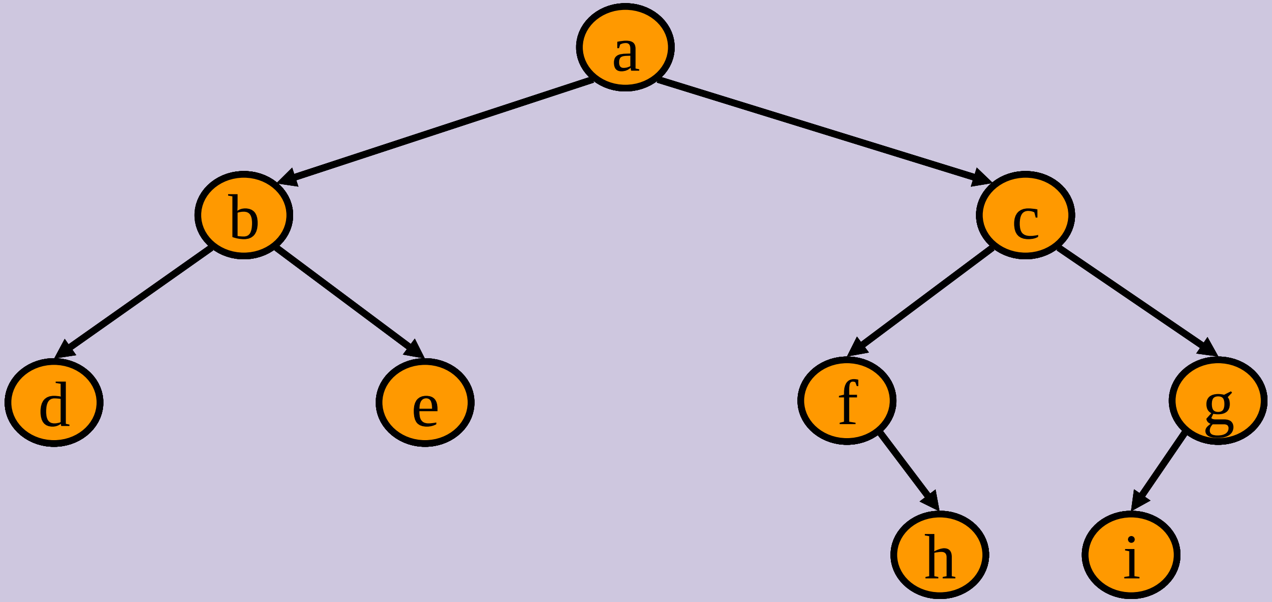
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

inorder

סייר בתת העץ השמאלי

בקר בשורש

סייר בתת העץ הימני

d b e a f h c i g

postorder

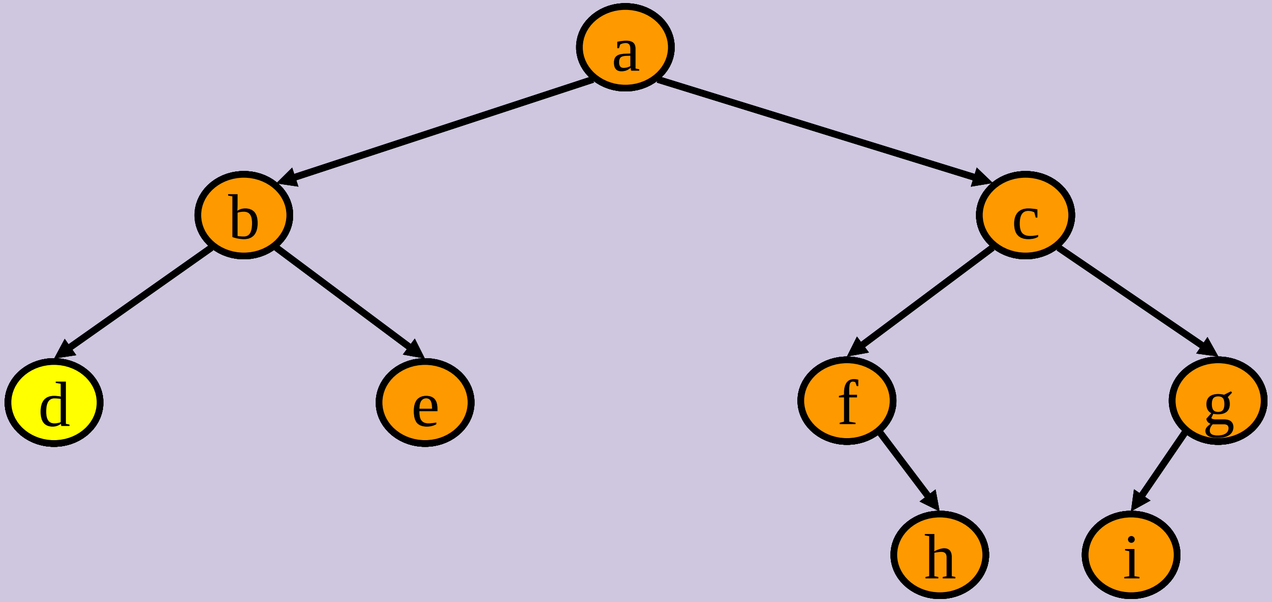
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

inorder

סייר בתת העץ השמאלי

בקר בשורש

סייר בתת העץ הימני

d b e a f h c i g

postorder

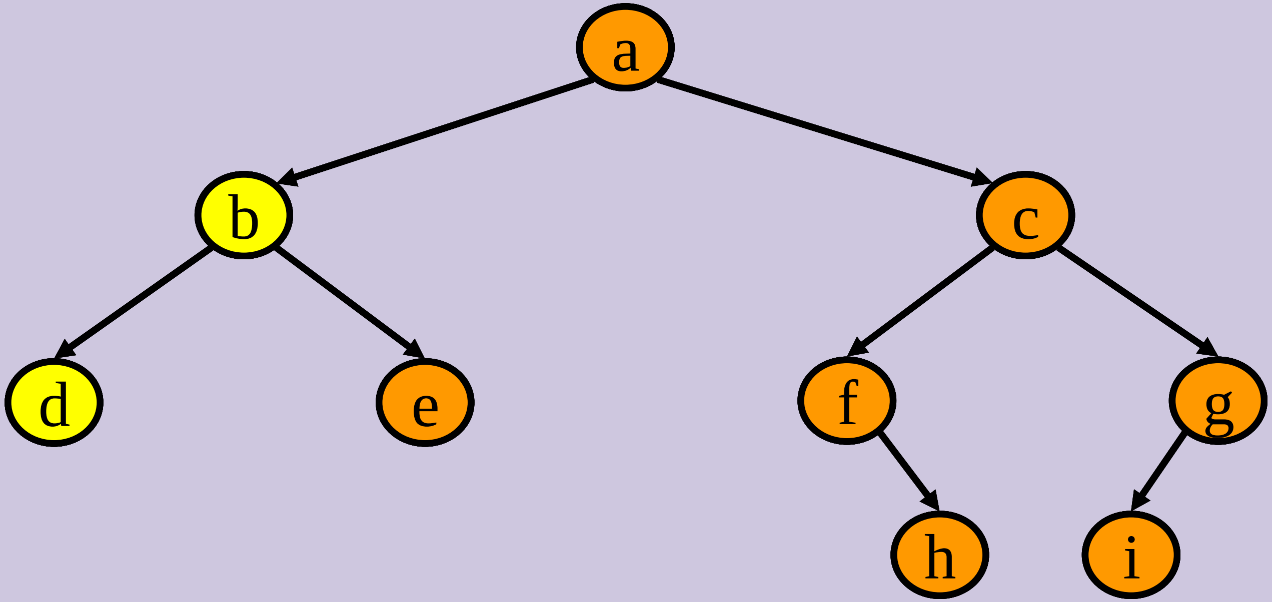
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

inorder

סייר בתת העץ השמאלי

בקר בשורש

סייר בתת העץ הימני

d b e a f h c i g

postorder

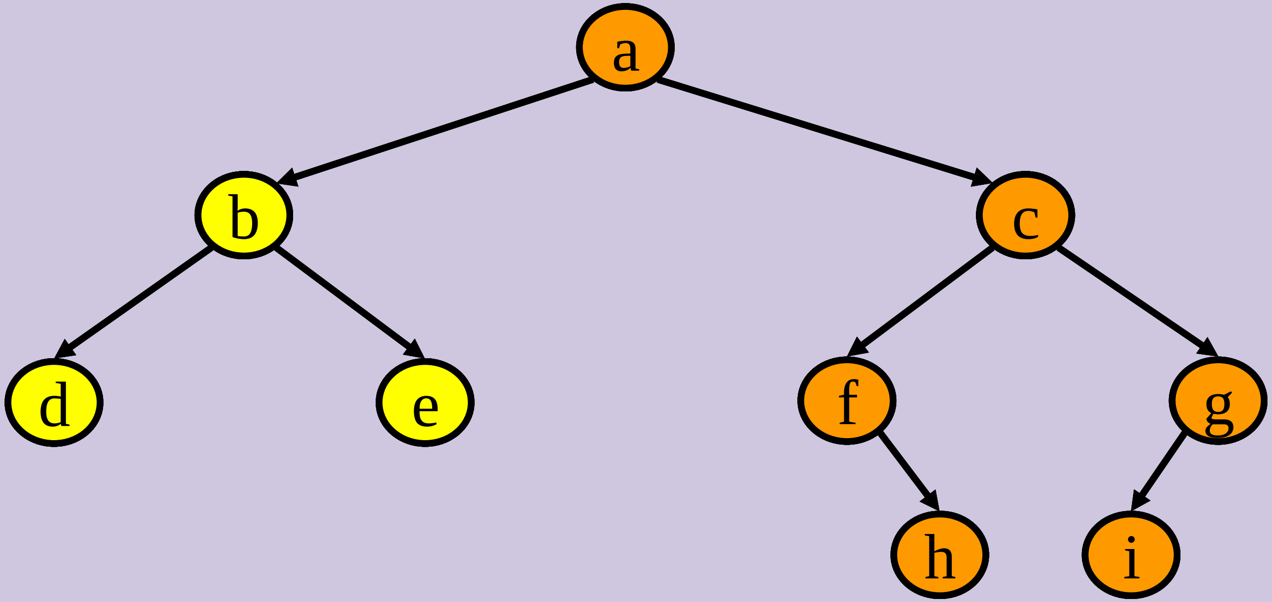
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

inorder

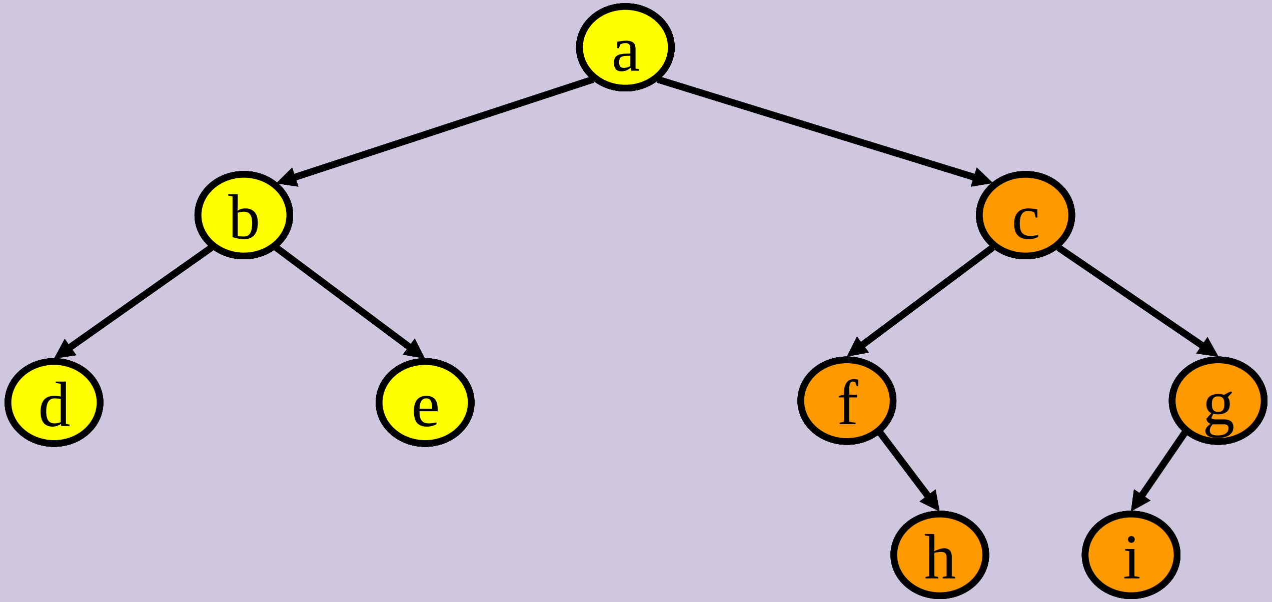
סייר בתת העץ השמאלי

בקר בשורש

סייר בתת העץ הימני

d b e a f h c i g

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

inorder

סייר בתת העץ השמאלי

בקר בשורש

סייר בתת העץ הימני

d b e a f h c i g

postorder

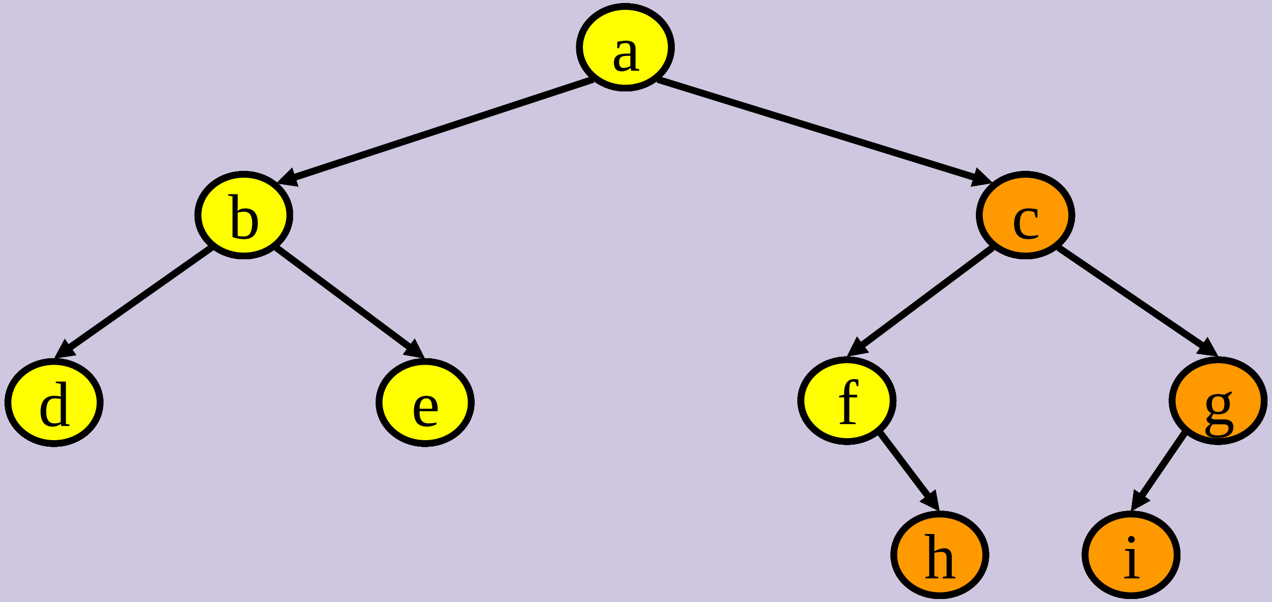
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

inorder

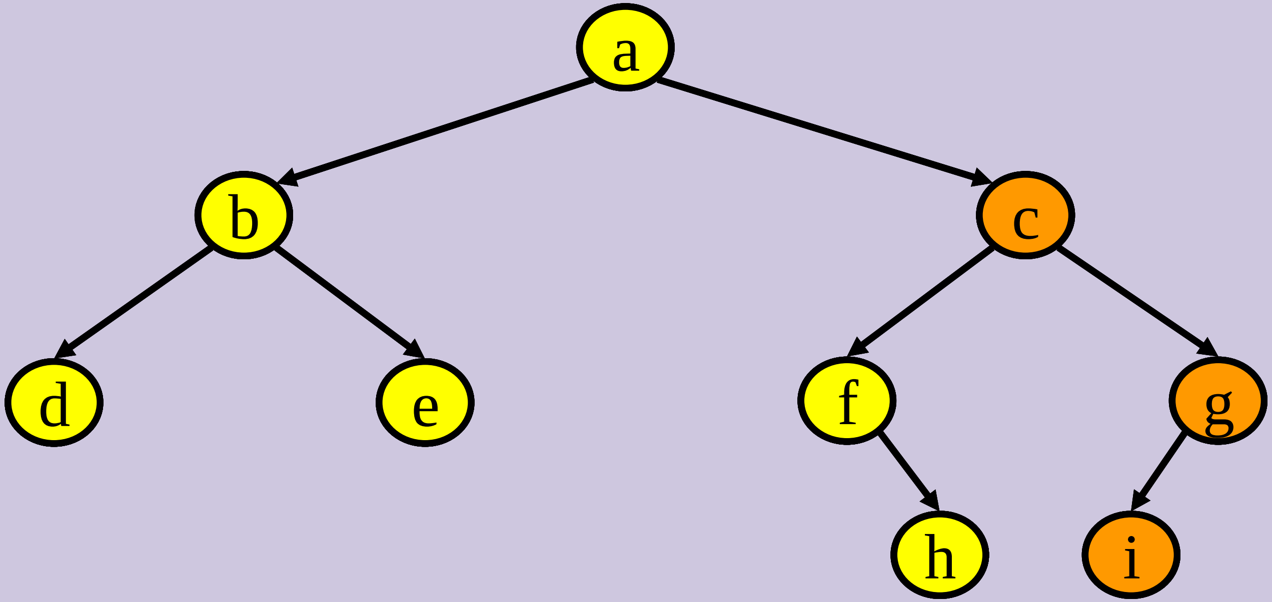
סייר בתת העץ השמאלי

בקר בשורש

סייר בתת העץ הימני

d b e a f h c i g

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

inorder

סייר בתת העץ השמאלי

בקר בשורש

סייר בתת העץ הימני

d b e a f h c i g

postorder

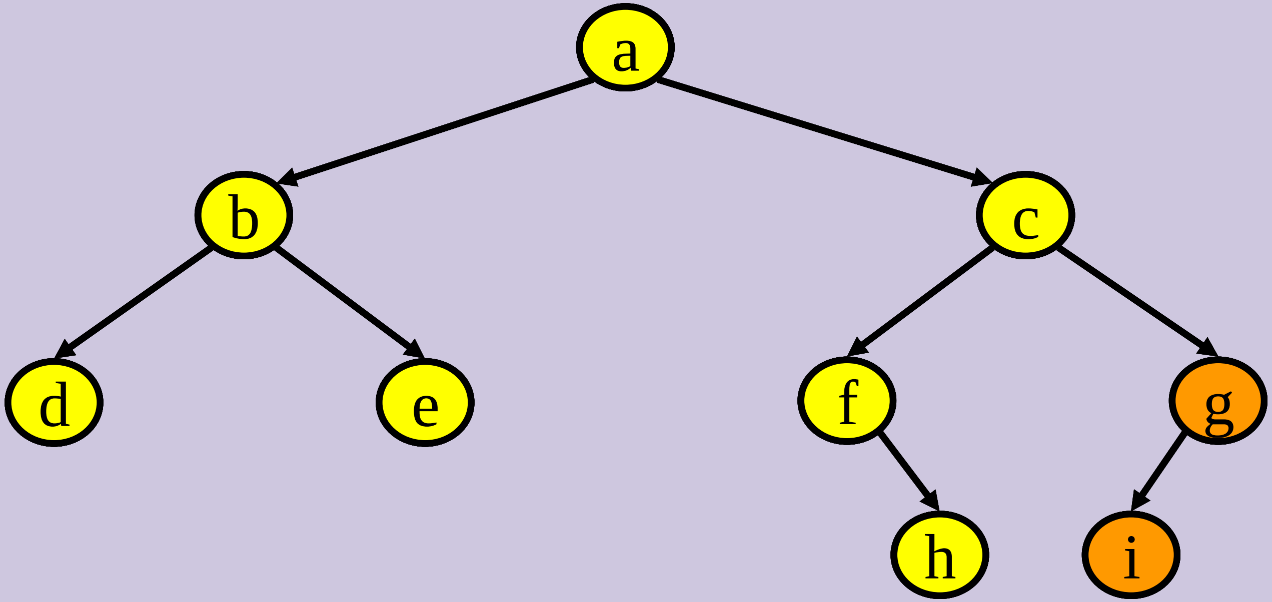
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

inorder

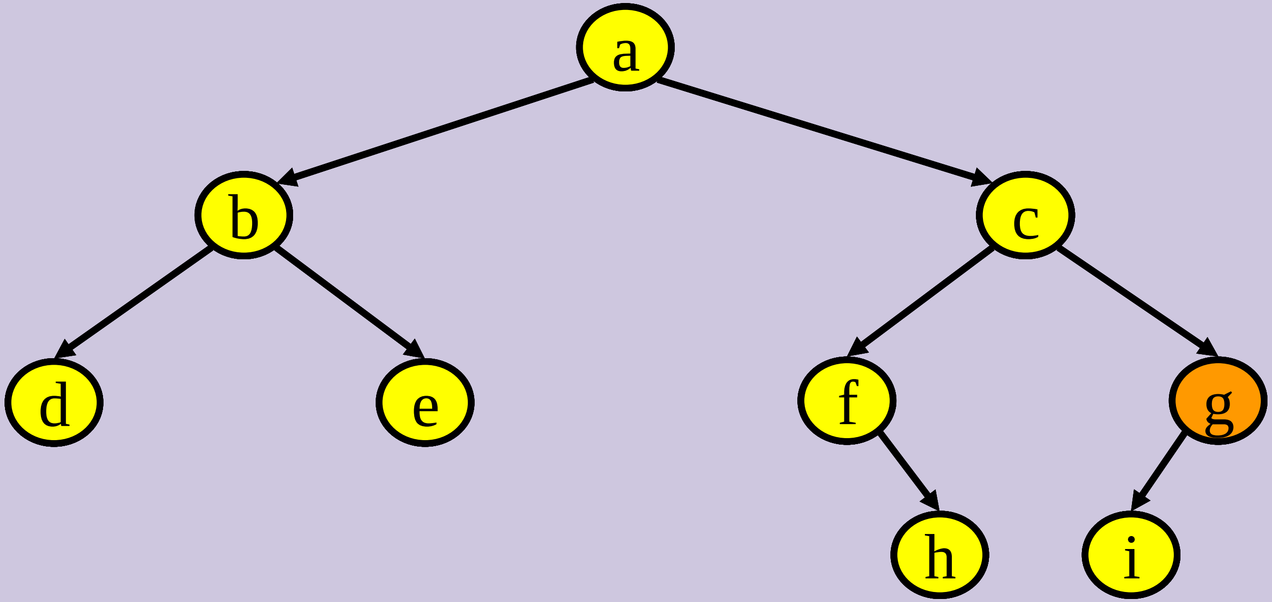
סייר בתת העץ השמאלי

בקר בשורש

סייר בתת העץ הימני

d b e a f h c i g

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

inorder

סייר בתת העץ השמאלי

בקר בשורש

סייר בתת העץ הימני

d b e a f h c i g

postorder

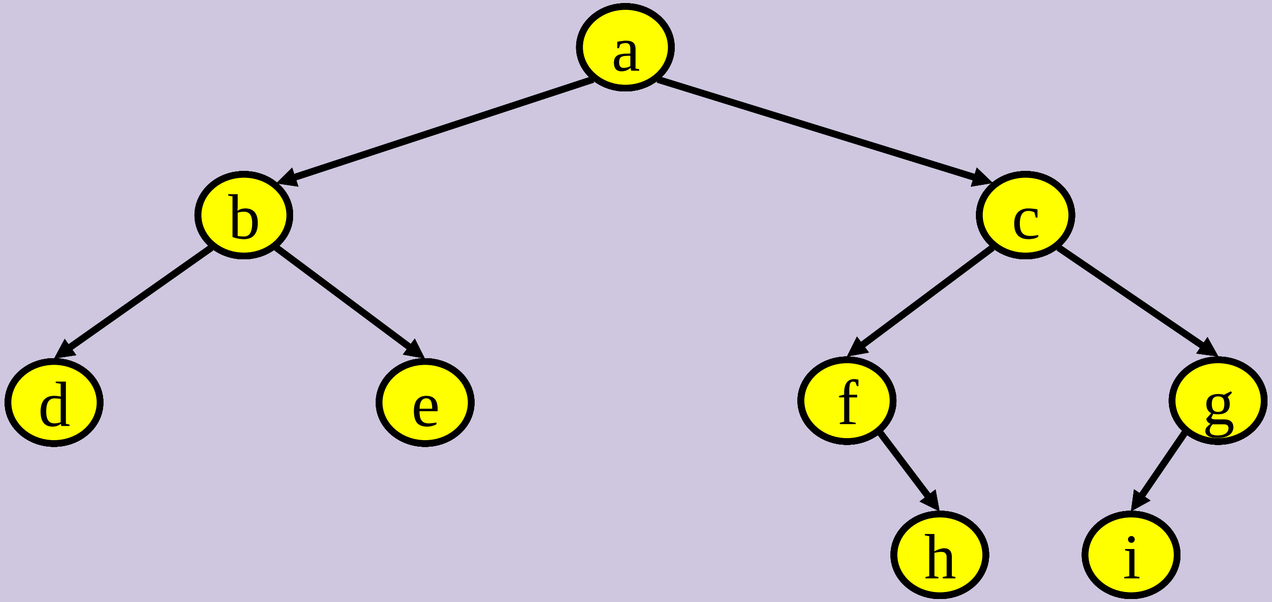
סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

סיור בעצים בינריים



preorder

בקר בשורש

סייר בתת העץ השמאלי

סייר בתת העץ הימני

a b d e c f h g i

postorder

סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשוב ביטויים אריתמטיים)

d e b h f i g c a

inorder

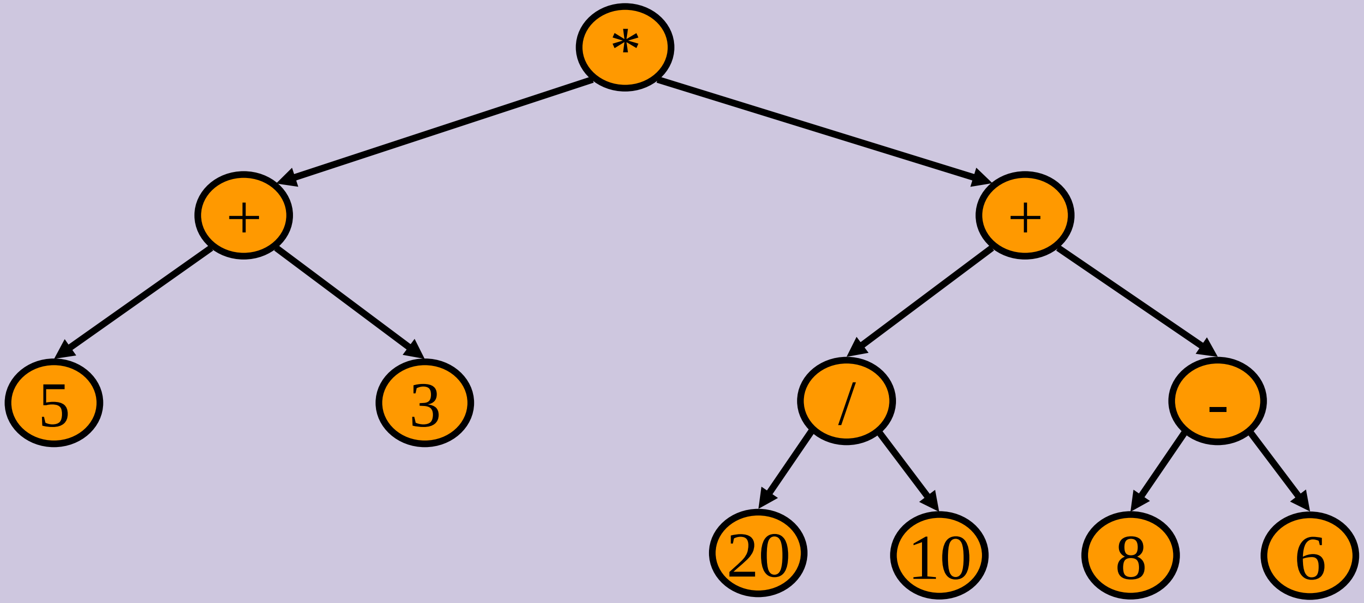
סייר בתת העץ השמאלי

בקר בשורש

סייר בתת העץ הימני

d b e a f h c i g

חשוב והמרה של ביטויים אריתמטיים



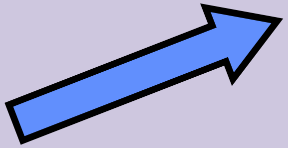
inorder

סייר בתת העץ השמאלי

בקר בשורש (הדפס)

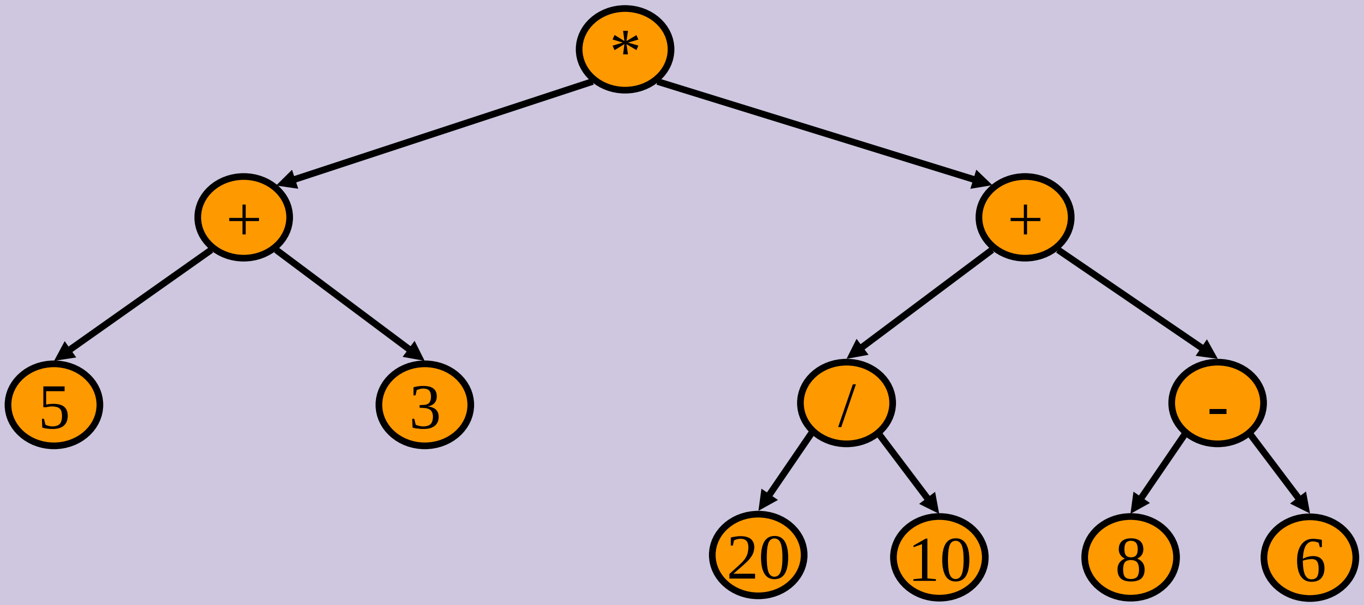
סייר בתת העץ הימני

(5+3)*((20/10)+(8-6))



infix

חשוב והמרה של ביטויים אריתמטיים



inorder

סייר בתת העץ השמאלי

בקר בשורש (הדפס)

סייר בתת העץ הימני

(5+3)*((20/10)+(8-6))

infix

postorder

סייר בתת העץ השמאלי

סייר בתת העץ הימני

בקר בשורש (חשב את ערך הביטוי או הדפס).

5 3 + 20 10 / 8 6 - + *

postfix

חישוב ביטוי postfix באמצעות מחסנית

אלגוריתם לחישוב ביטוי postfix

1. התחל עם מחסנית ריקה.
2. עבור על הביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד- הכנס אותו למחסנית.
4. אם הוא פעולה – הפעל את הפעולה על שני האיברים
שבראש המחסנית והכנס את התוצאה למחסנית.

postfix: 5 3 + 20 10 / 8 6 - + *

חישוב ביטוי postfix באמצעות מחסנית

אלגוריתם לחישוב ביטוי postfix

1. התחל עם מחסנית ריקה.
2. עבור על הביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד- הכנס אותו למחסנית.
4. אם הוא פעולה – הפעל את הפעולה על שני האיברים שבראש המחסנית והכנס את התוצאה למחסנית.

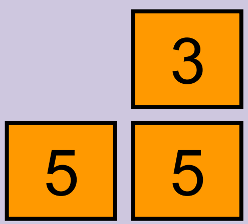
$\boxed{5}$

postfix: 5 3 + 20 10 / 8 6 - + *

חישוב ביטוי postfix באמצעות מחסנית

אלגוריתם לחישוב ביטוי postfix

1. התחל עם מחסנית ריקה.
2. עבור על הביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד- הכנס אותו למחסנית.
4. אם הוא פעולה – הפעל את הפעולה על שני האיברים
   שבראש המחסנית והכנס את התוצאה למחסנית.



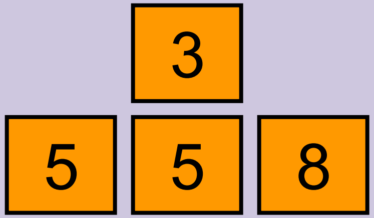
postfix: 5 3 + 20 10 / 8 6 - + *

חישוב ביטוי postfix באמצעות מחסנית

<u>אלגוריתם לחישוב ביטוי postfix</u>

1. התחל עם מחסנית ריקה.
2. עבור על הביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד- הכנס אותו למחסנית.
4. אם הוא פעולה – הפעל את הפעולה על שני האיברים

   שבראש המחסנית והכנס את התוצאה למחסנית.

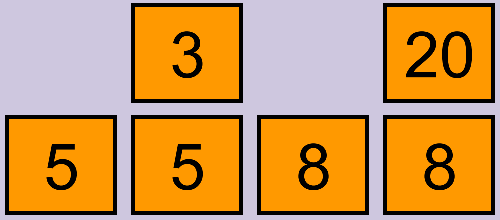


5 3 + 20 10 / 8 6 - + * :postfix

חישוב ביטוי postfix באמצעות מחסנית

<u>אלגוריתם לחישוב ביטוי postfix</u>

1. התחל עם מחסנית ריקה.
2. עבור על הביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד- הכנס אותו למחסנית.
4. אם הוא פעולה – הפעל את הפעולה על שני האיברים שבראש המחסנית והכנס את התוצאה למחסנית.

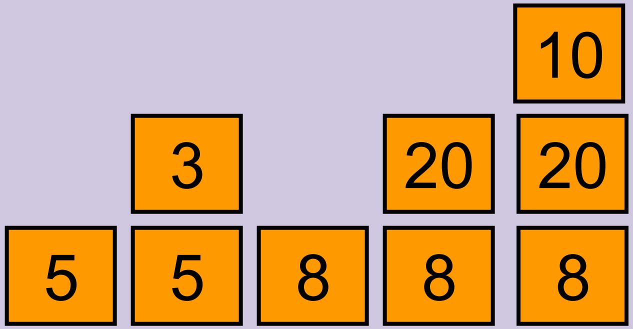


postfix: 5 3 + 20 10 / 8 6 - + *

חישוב ביטוי postfix באמצעות מחסנית

אלגוריתם לחישוב ביטוי postfix

1. התחל עם מחסנית ריקה.
2. עבור על הביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד- הכנס אותו למחסנית.
4. אם הוא פעולה – הפעל את הפעולה על שני האיברים שבראש המחסנית והכנס את התוצאה למחסנית.



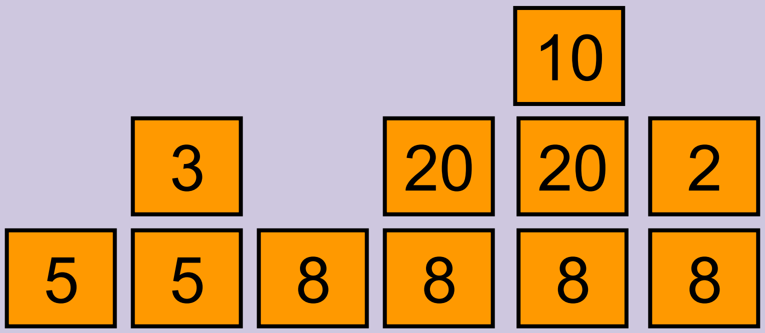
postfix: 5 3 + 20 10 / 8 6 - + *

חישוב ביטוי postfix באמצעות מחסנית

<u>אלגוריתם לחישוב ביטוי postfix</u>

1. התחל עם מחסנית ריקה.
2. עבור על הביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד- הכנס אותו למחסנית.
4. אם הוא פעולה – הפעל את הפעולה על שני האיברים

שבראש המחסנית והכנס את התוצאה למחסנית.

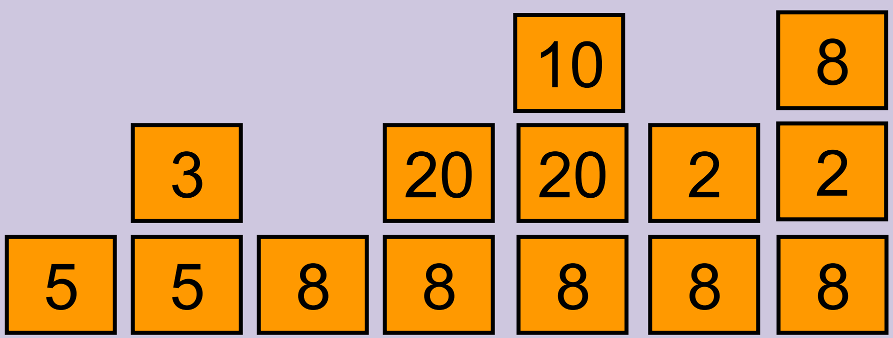


postfix: 5 3 + 20 10 / 8 6 - + *

חישוב ביטוי postfix באמצעות מחסנית

<u>אלגוריתם לחישוב ביטוי postfix</u>

1. התחל עם מחסנית ריקה.
2. עבור על הביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד- הכנס אותו למחסנית.
4. אם הוא פעולה – הפעל את הפעולה על שני האיברים שבראש המחסנית והכנס את התוצאה למחסנית.



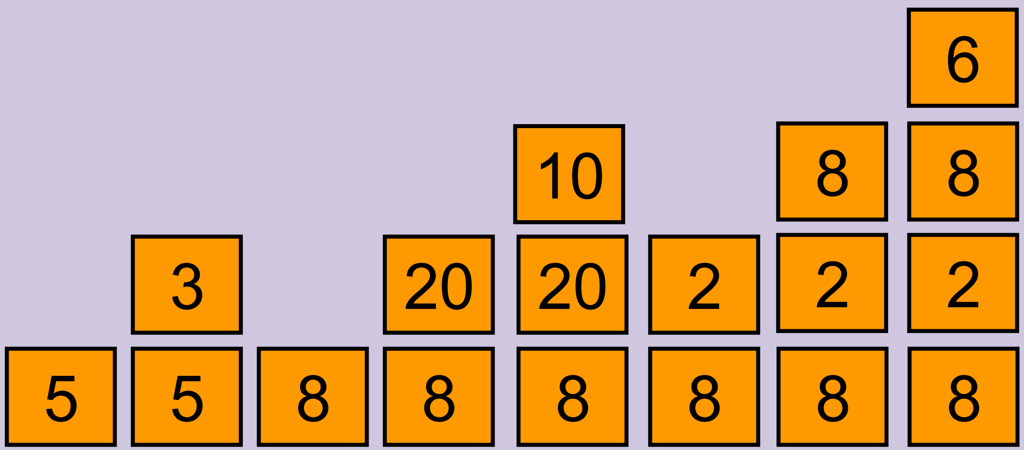
postfix: 5 3 + 20 10 / 8 6 - + *

חישוב ביטוי postfix באמצעות מחסנית

<u>אלגוריתם לחישוב ביטוי postfix</u>

1. התחל עם מחסנית ריקה.
2. עבור על הביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד- הכנס אותו למחסנית.
4. אם הוא פעולה – הפעל את הפעולה על שני האיברים

שבראש המחסנית והכנס את התוצאה למחסנית.



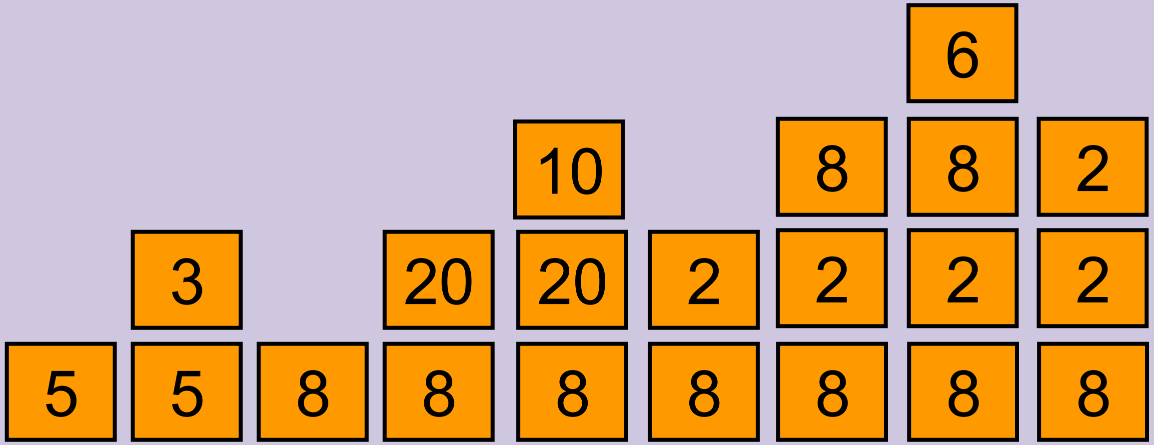
postfix: 5 3 + 20 10 / 8 6 - + *

חישוב ביטוי postfix באמצעות מחסנית

<u>אלגוריתם לחישוב ביטוי postfix</u>

1. התחל עם מחסנית ריקה.
2. עבור על הביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד- הכנס אותו למחסנית.
4. אם הוא פעולה – הפעל את הפעולה על שני האיברים

   שבראש המחסנית והכנס את התוצאה למחסנית.

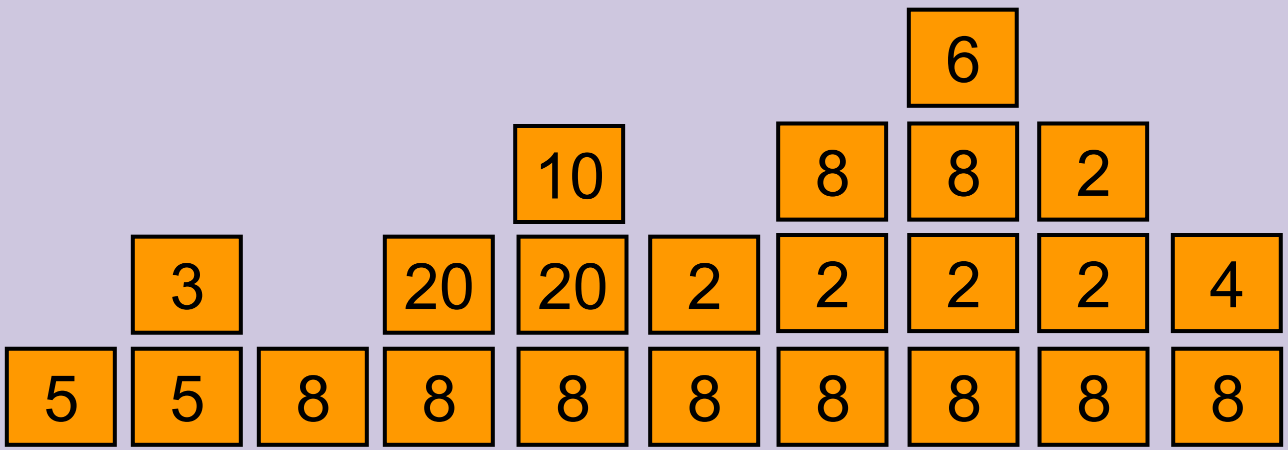


postfix: 5 3 + 20 10 / 8 6 - + *

חישוב ביטוי postfix באמצעות מחסנית

<u>אלגוריתם לחישוב ביטוי postfix</u>

1. התחל עם מחסנית ריקה.
2. עבור על הביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד- הכנס אותו למחסנית.
4. אם הוא פעולה – הפעל את הפעולה על שני האיברים שבראש המחסנית והכנס את התוצאה למחסנית.



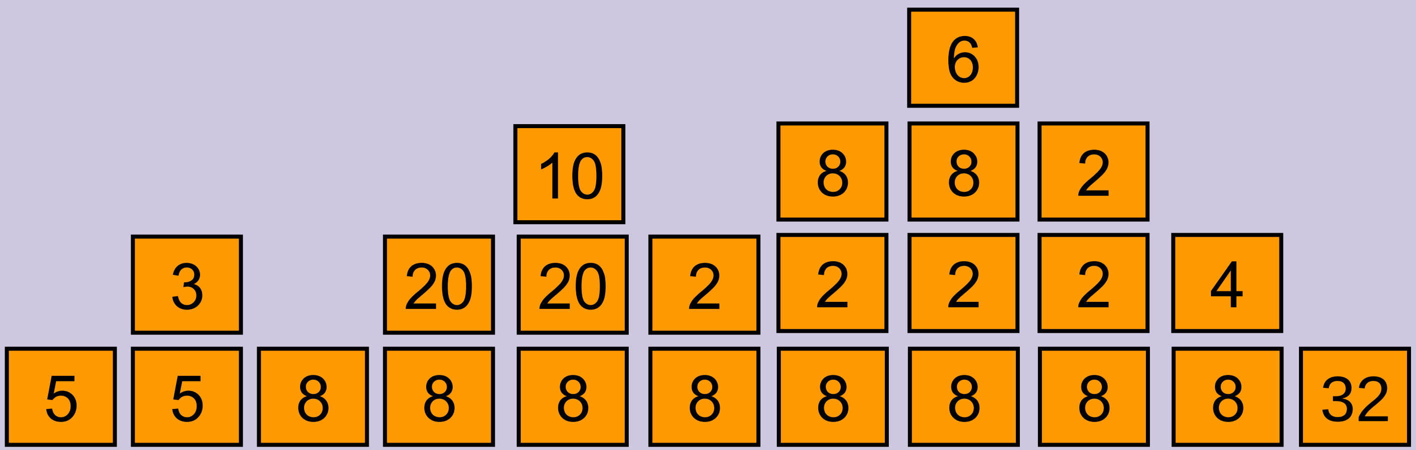
postfix: 5 3 + 20 10 / 8 6 - + *

חישוב ביטוי postfix באמצעות מחסנית

אלגוריתם לחישוב ביטוי postfix

1. התחל עם מחסנית ריקה.
2. עבור על הביטוי משמאל לימין:
3. אם האיבר הבא הוא אופרנד- הכנס אותו למחסנית.
4. אם הוא פעולה – הפעל את הפעולה על שני האיברים

שבראש המחסנית והכנס את התוצאה למחסנית.



postfix: 5 3 + 20 10 / 8 6 - + *

מימוש פרוצדורת postorder

```
void postorder (NODE *T)
{
if (T = = NULL) return;
else {
        postorder( T → left);        /* 1*/
        postorder( T → right);       /* 2*/
        “visit”;                     /* 3*/
        return;}
}
```

מבנה הצומת:

value	
left	right

```
typedef struct node {
  int value;
  struct node *left, *right;
   }  NODE;
```

מימוש פרוצדורת postorder

```
void postorder (NODE *T)
{
if (T = = NULL) return;
else {
        postorder( T → left);      /* 1*/
        postorder( T → right);     /* 2*/
        “visit”;                   /* 3*/
        return;}
}
```

מבנה הצומת:

value	
left	right

```
typedef struct node {
  int value;
  struct node *left, *right;
   }  NODE;
```

<u>הערה</u>: ע"י החלפת שורה #2 עם שורה #3 נקבל מימוש של סיור inorder.

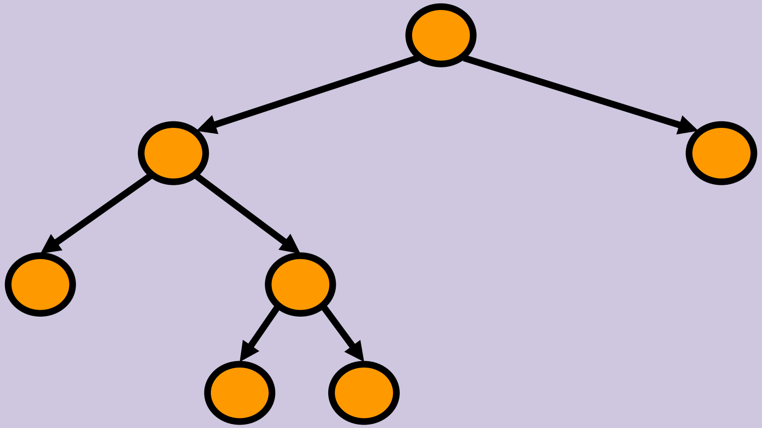
<u>תרגיל I</u>: יש לשנות תוכנית זו כך שתחשב ערך של ביטוי אריתמטי.

<u>תרגיל II</u>: יש לכתוב תוכנית זו ללא רקורסיה (תוך שימוש במחסנית).

מימוש עצים בינריים

פונקציה רקורסיבית לחישוב גובה העץ (דוגמא לסיור postorder):

```
int height (NODE *T)
{         int L,R;
          if (T = = NULL)  return  –1
          else {
                    L = height(T → left);
                    R = height(T → right);
                    return 1 + max(L,R) ;
               }
}
```



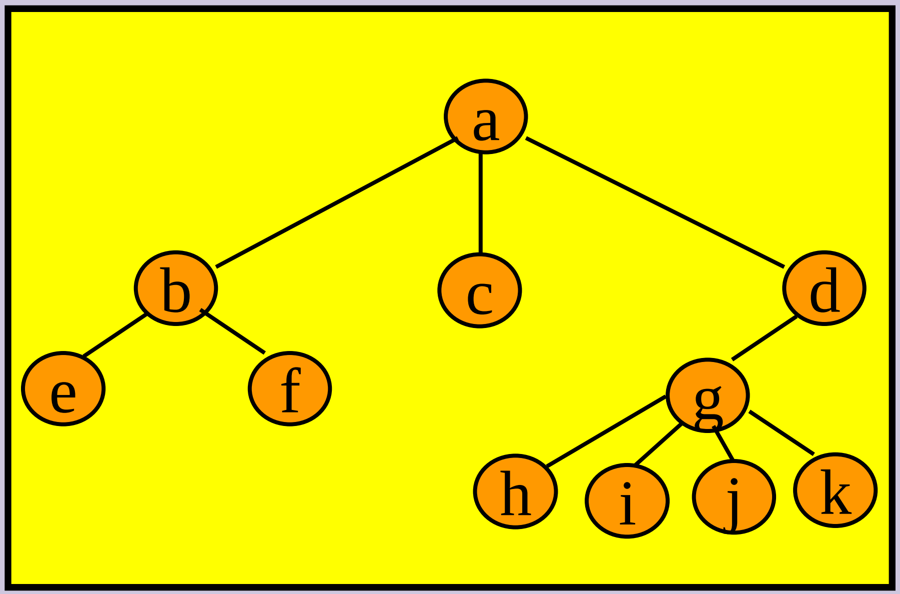
עצים מסודרים

אם לכל צמת דרגה ≤ d:

value(s)			
child[0]	child[1]	…	child[d-1]

נתן לייצג עץ מדרגה כלשהי בצורה הבאה:

value(s)	
first-child	next-brother



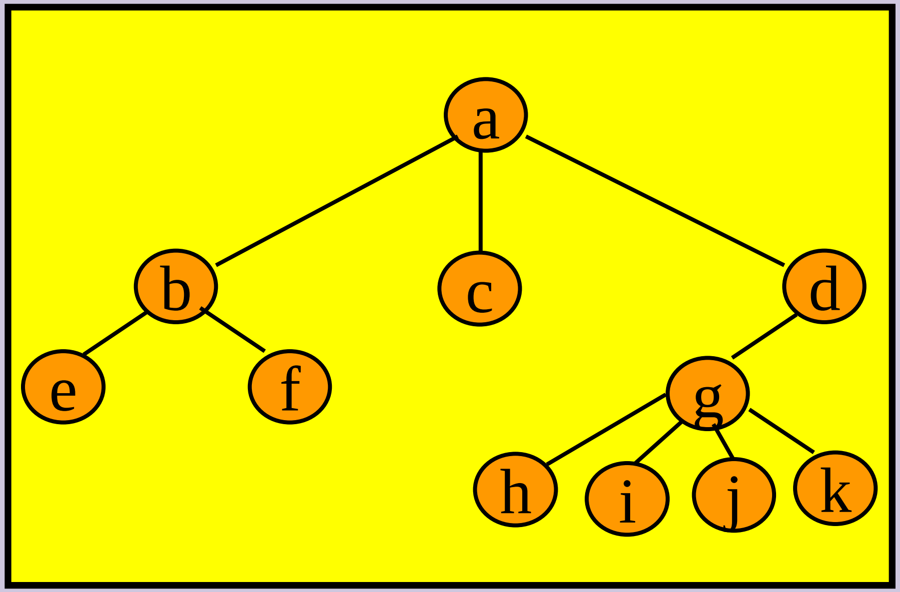
עצים מסודרים

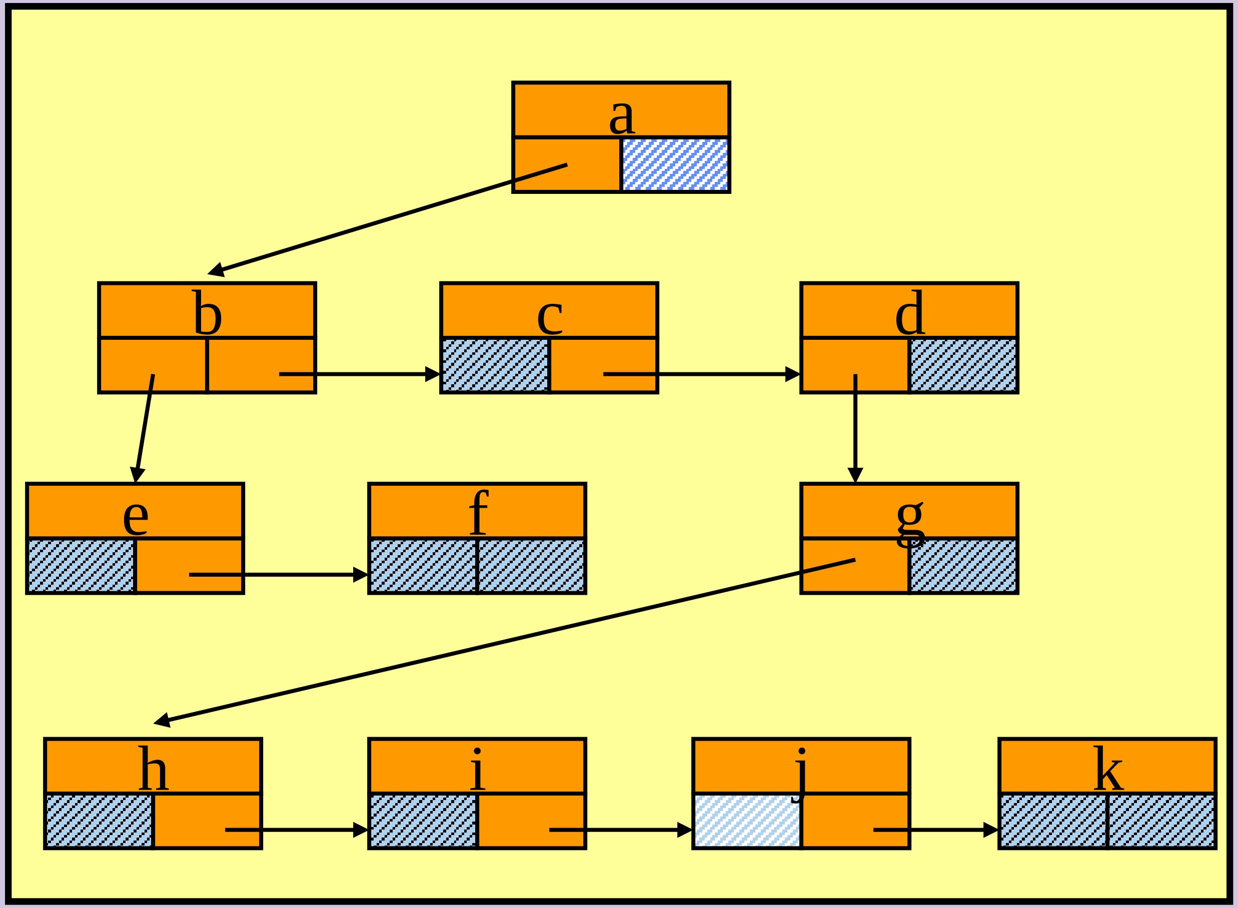
אם לכל צמת דרגה $\leq d$:

value(s)			
child[0]	child[1]	…	child[d-1]

נתן לייצג עץ מדרגה כלשהי בצורה הבאה:

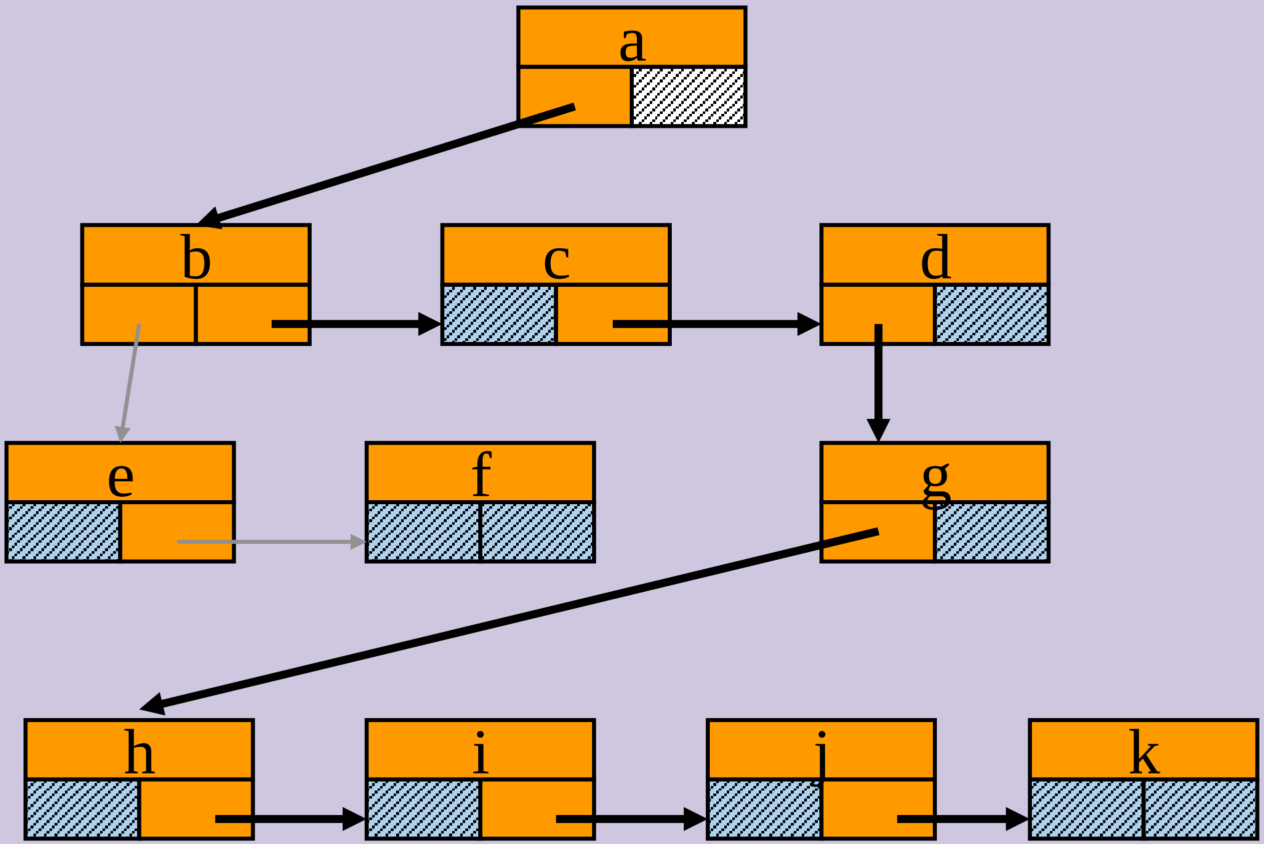
value(s)	
first-child	next-brother





עצים מסודרים

מה הקשר בין גובה העץ המקורי וגובה העץ הבינרי ?



$h_{new} \le d\ h_{old}$ כאשר d הוא מספר הבנים המקסימלי בעץ המקורי.

מילון (Dictionary)

מילון מאחסן אוסף של רשומות מהטיפוס (מפתח, אינפורמציה). המפתח שונה (בד"כ) מרשומה לרשומה. אוסף המפתחות האפשריים מסומן ב-U.

מפתחות אפשריים לדוגמא: מספרים שלמים.

<u>פעולות:</u>

- אתחול. יצירת מילון ריק. create(D)
- חיפוש. החזר מצביע לרשומה ב- D שמפתחה x או NULL. find(D,x)
- הוספה. הוסף ל- D רשומה שמפתחה x. insert(D,x,info)
- הוצאה. סלק מ- D רשומה שמפתחה x. delete(D,x)

<u>כללים:</u>

- x שייך לקבוצת המפתחות U.
- כל x מופיע לכל היותר פעם אחת במילון (בדר"כ).

מילון ועצי חיפוש

<u>פעולות נוספות כאשר מוגדר סדר על U (למשל כאשר מפתח הוא מספר):</u>

min(D) מינימום. החזר את המפתח המינימלי ב- D.

next(D,x) עוקב. החזר מצביע לאיבר במילון D בעל המפתח הקטן ביותר שגדול מ- x.

<u>מטרה</u>: לבצע את כל הפעולות בזמן O(log n) (במקרה הגרוע ביותר) כאשר n הוא מספר המפתחות הנמצאים במילון <u>בזמן ביצוע הפעולה</u>.

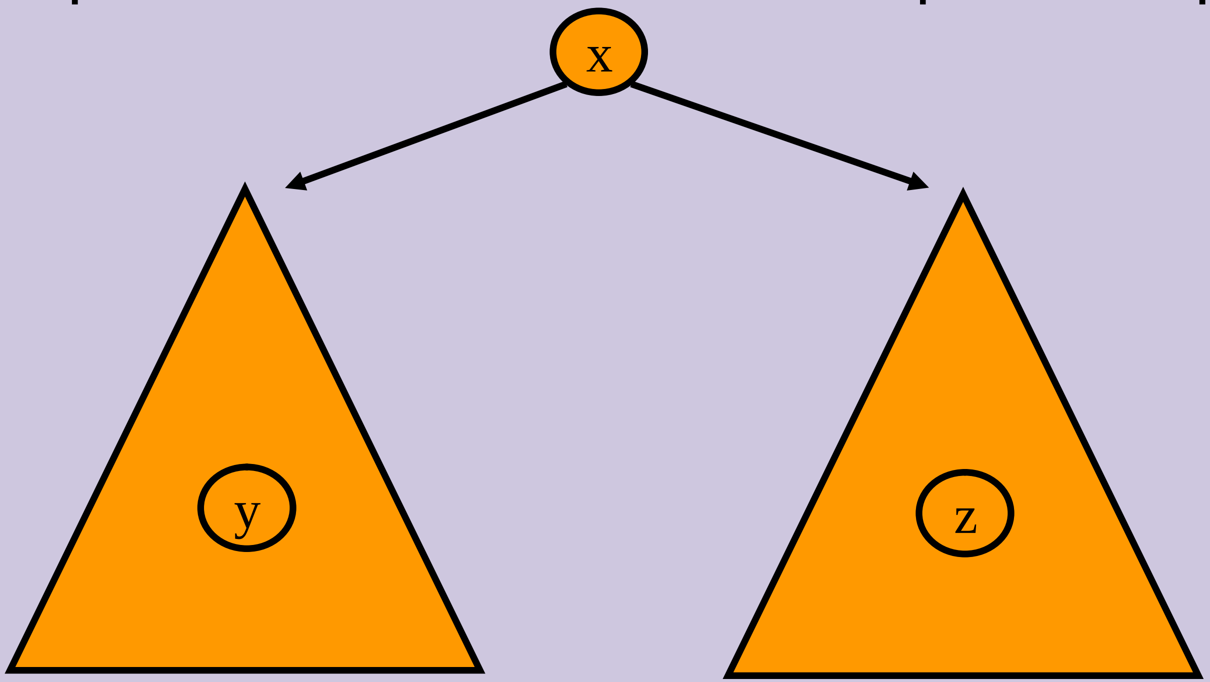
<u>עצי חיפוש</u> היא משפחה של מימושים למבנה הנתונים "מילון" כאשר מוגדר סדר על קבוצת המפתחות U.

<u>עץ בינרי כעץ חיפוש</u>

נשתמש בעץ בינרי מכוון.

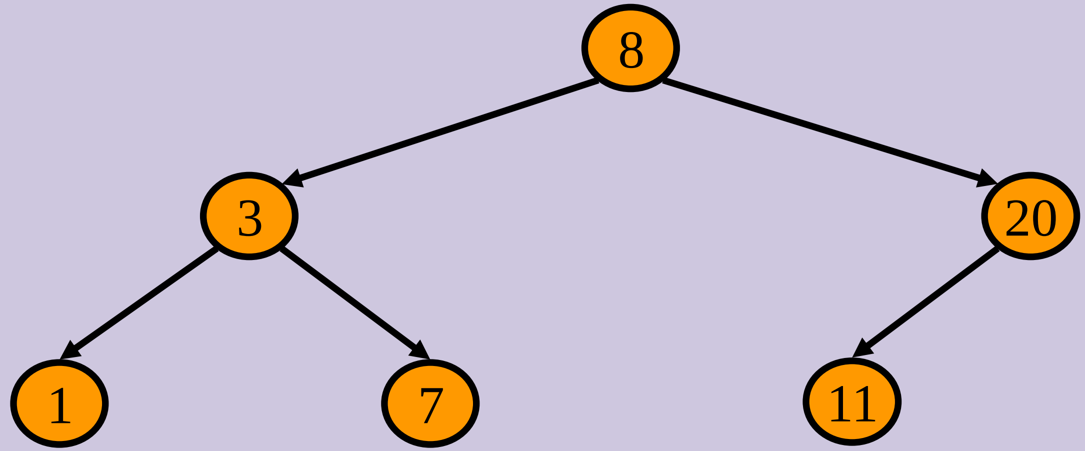
בכל צומת נאחסן רשומה אחת מתוך המילון (או מפתח ו<u>מצביע</u> לאינפורמציה של הרשומה).

נשמור על הכלל הבא: עבור צומת כלשהו בעל מפתח x, כל המפתחות בתת העץ השמאלי קטנים מ- x וכל המפתחות בתת העץ הימני גדולים מ- x.



$$y < x < z$$

עץ בינרי כעץ חיפוש

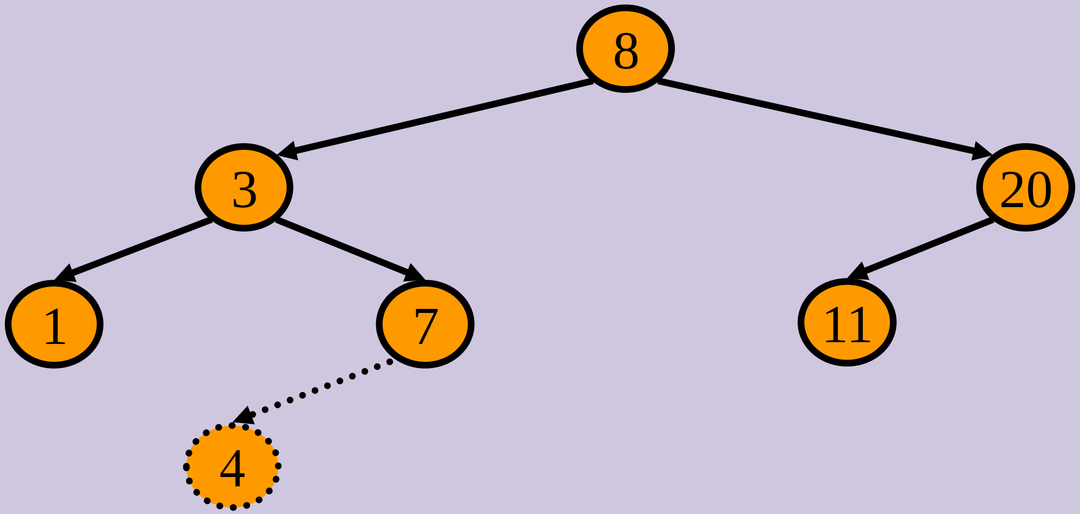


הערה: בציור מופיעים רק המפתחות ולא הרשומות במלואן.

אלגוריתם החיפוש: find(T,x)

1. אם T ריק, דווח ש-x לא בעץ.
2. יהי y הערך שבשורש.
3. אם y = x, החזר מצביע לצומת המחזיק את x.
4. אם y< x, המשך את החיפוש בתת העץ השמאלי של T.
5. אחרת (כאשר y > x), המשך את החיפוש בתת העץ הימני של T.

הכנסה בעץ חיפוש

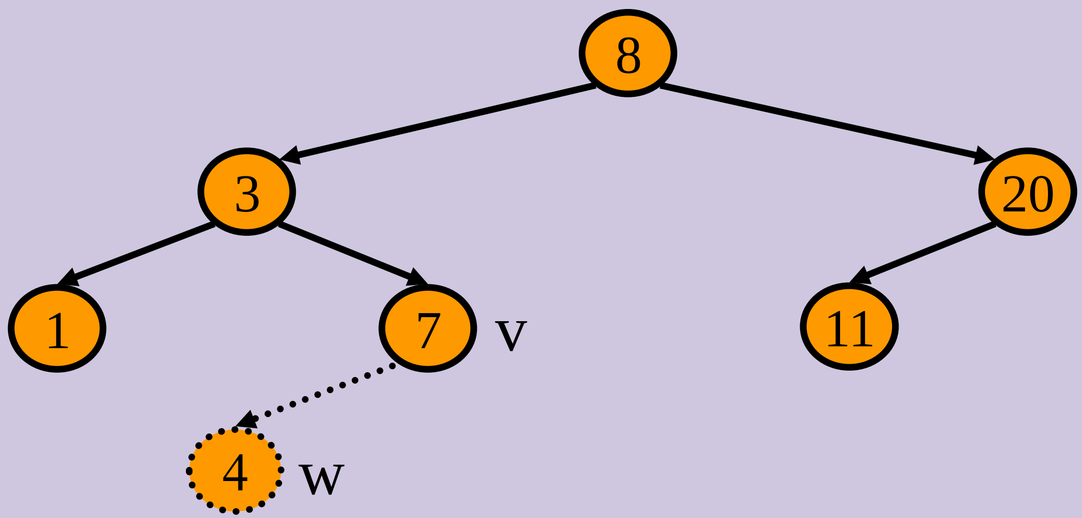


insert(T,4)

הכנסה בעץ חיפוש

אלגוריתם הכנסה: insert(T,x)

1. חפש את x בעץ החיפוש T.
2. אם x נמצא ב-T, עצור ודווח.
3. יהי v הצומת האחרון במסלול החיפוש של x ויהי y המפתח שנמצא ב-v.

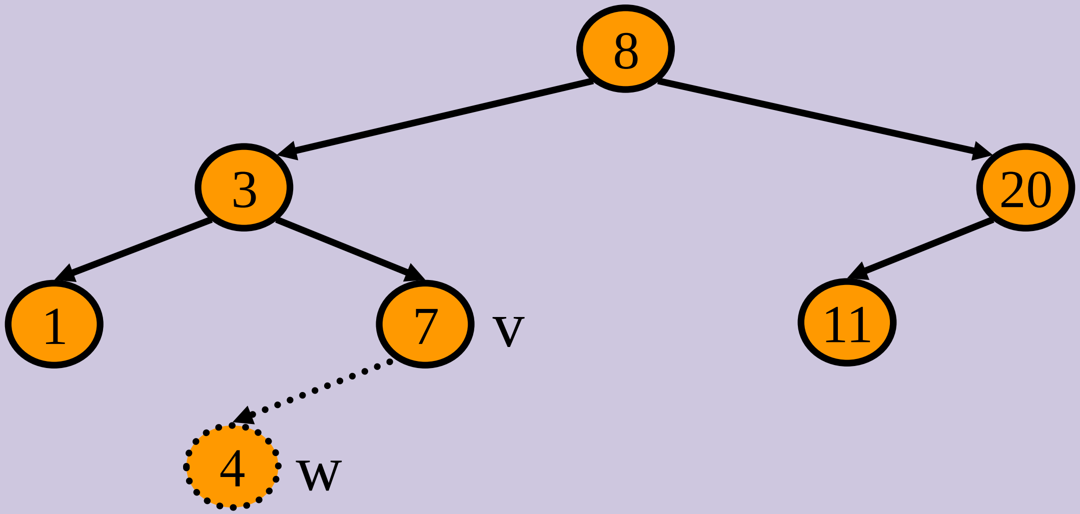


insert(T,4)

הכנסה בעץ חיפוש

<u>אלגוריתם הכנסה: insert(T,x)</u>

1. חפש את x בעץ החיפוש T.
2. אם x נמצא ב-T, עצור ודווח.
3. יהי v הצומת האחרון במסלול החיפוש של x ויהי y המפתח שנמצא ב-v.
4. אם x<y, הוסף צומת w עם מפתח x כבן שמאלי של v.
5. אחרת (כאשר x > y), הוסף צומת w עם מפתח x כבן ימני של v.

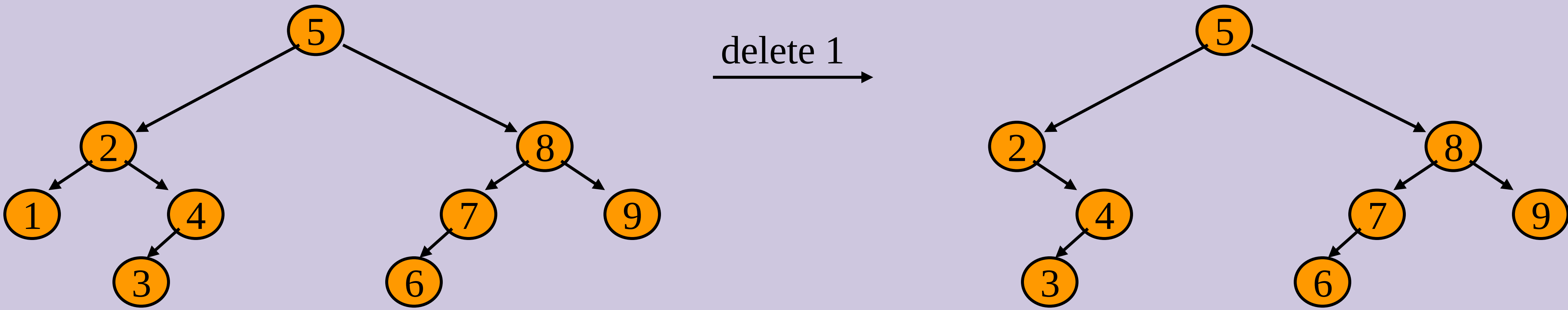


insert(T,4)

הוצאה מעץ חיפוש

אלגוריתם הוצאה. יהי v צומת בעץ המיועד להוצאה.

1. אם v עלה, סלק אותו.
2. אם ל-v בן יחיד, תן לאבא של v להצביע על הבן.

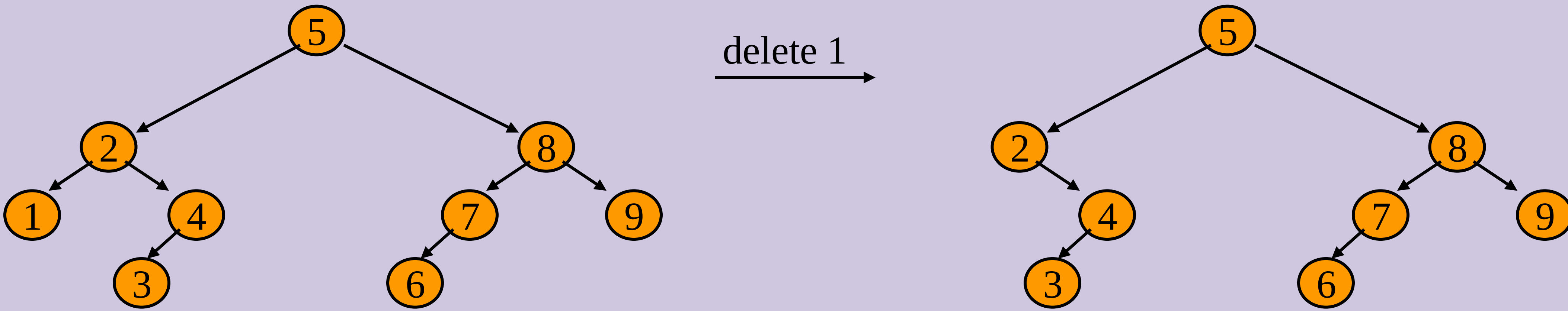


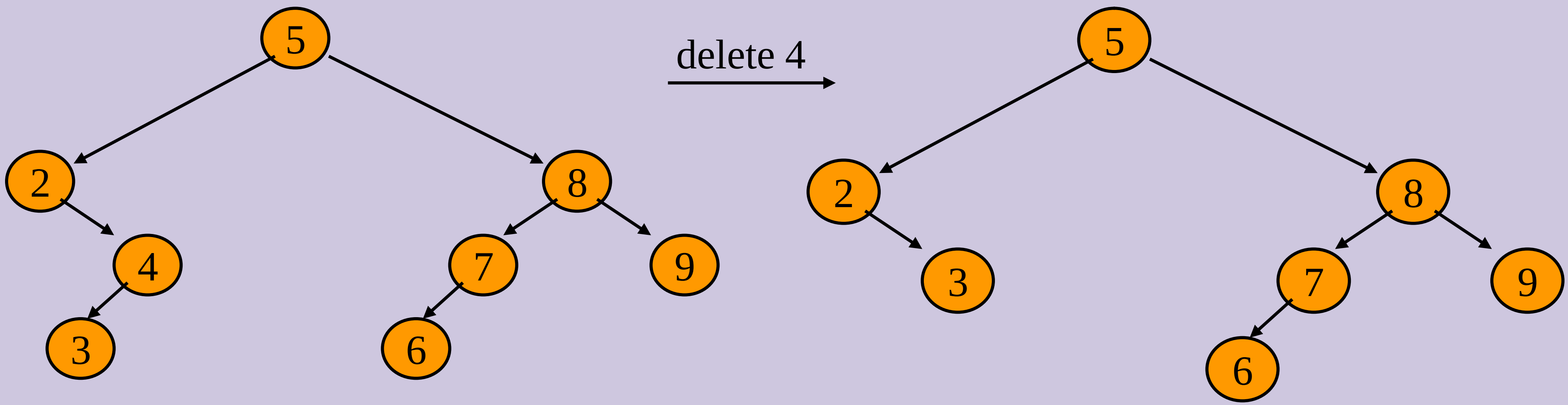
הוצאה מעץ חיפוש

אלגוריתם הוצאה. יהי v צומת בעץ המיועד להוצאה.

1. אם v עלה, סלק אותו.

2. אם ל-v בן יחיד, תן לאבא של v להצביע על הבן.

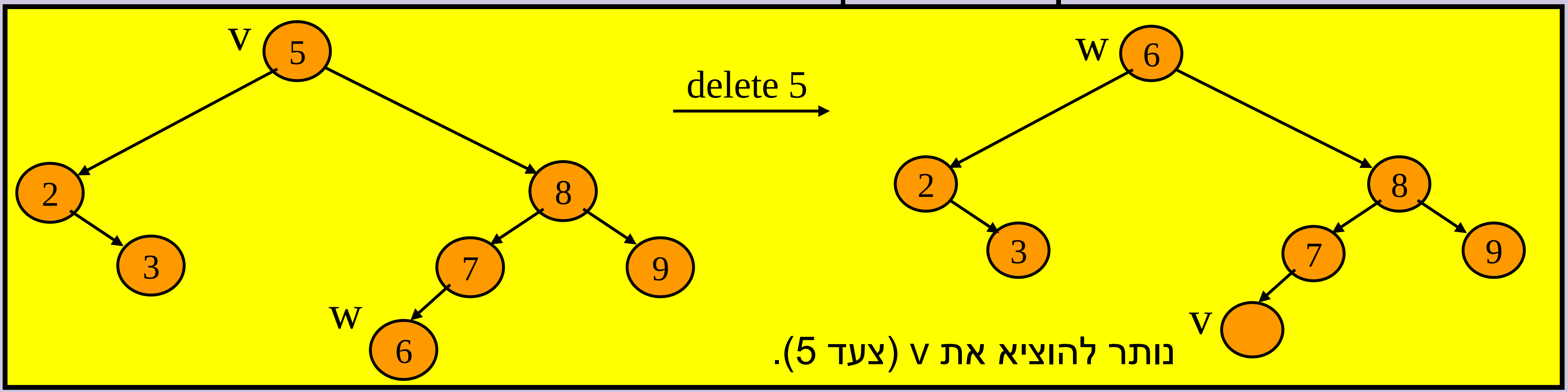




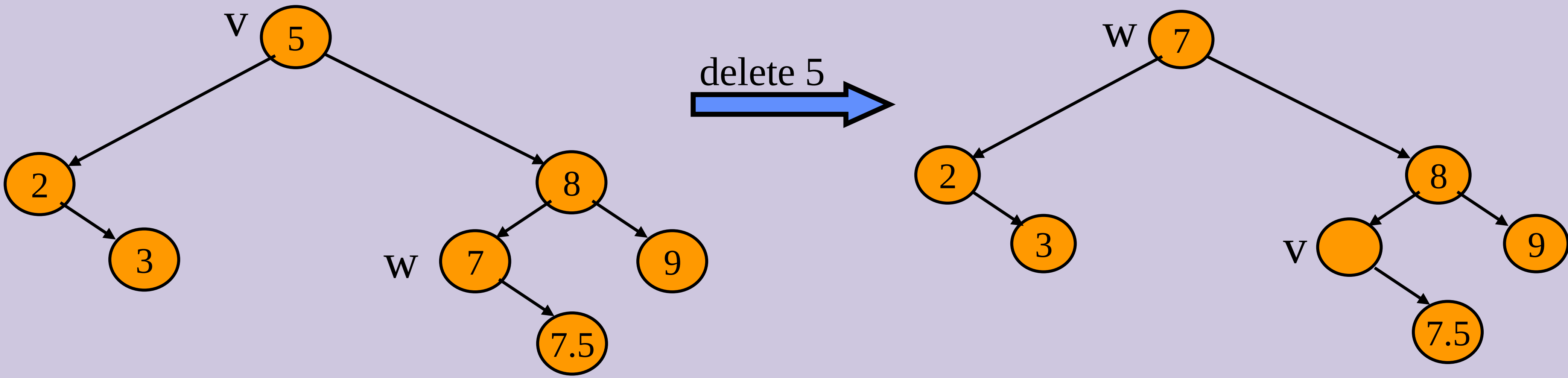
הוצאה מעץ חיפוש

אלגוריתם הוצאה. יהי v צומת בעץ המיועד להוצאה.

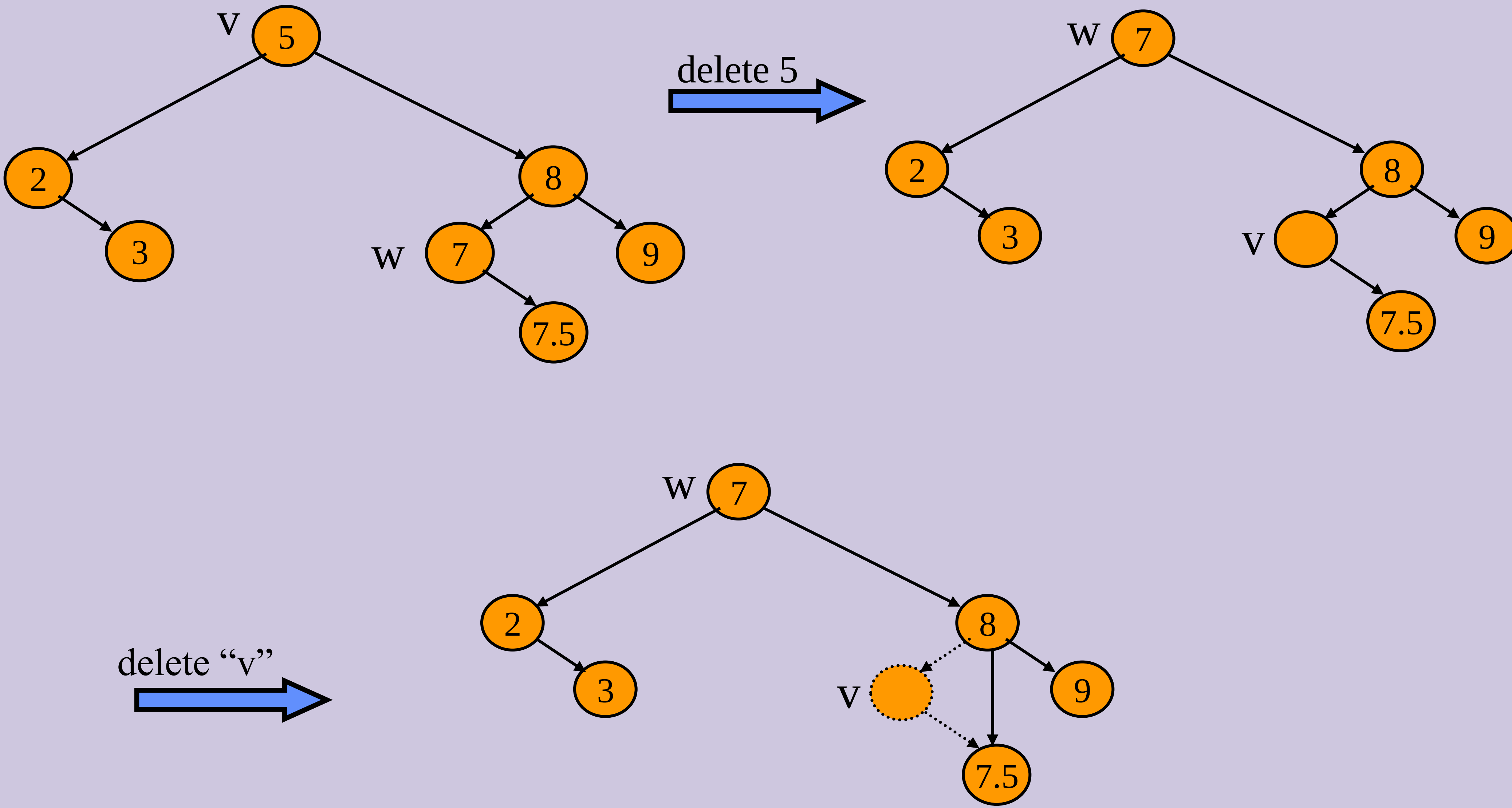
1. אם v עלה, סלק אותו.
2. אם ל-v בן יחיד, תן לאבא של v להצביע על הבן.
3. אחרת: יהי w הצומת העוקב ל-v בסדר inorder. (זהו הצומת המכיל את הערך הבא אחרי הערך שב- v כלומר הצומת המתקבל ע"י פניה אחת ימינה ואח"כ כל הדרך שמאלה. שימו לב שלצומת w בן אחד לכל היותר).
4. החלף בין צומת v וצומת w.
5. כעת יש ל-v לכל היותר בן אחד. המשך בצעד 1 או 2 כנדרש.



דוגמא נוספת



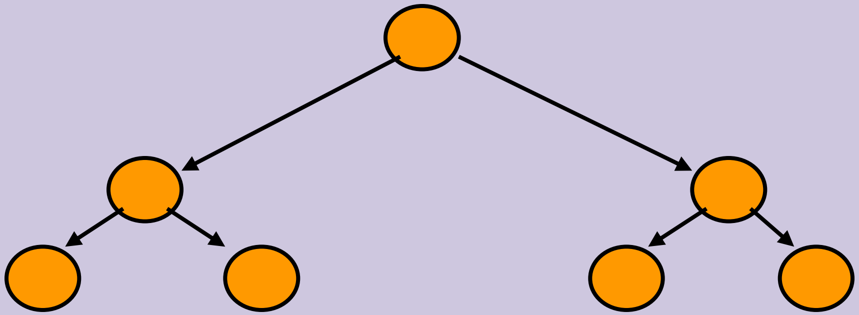
דוגמא נוספת



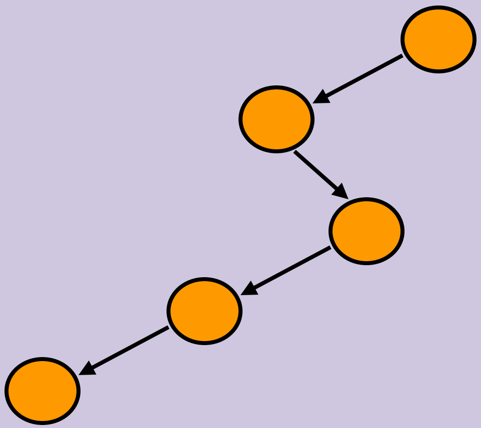
נתוח זמנים

זמן חיפוש/הכנסה/הוצאה הוא לינארי בגובה העץ.

מהו גובה העץ?



מקרה טוב. עץ שלם. $h=\lfloor \log n \rfloor$



מקרה גרוע. עץ הנראה כרשימה ליניארית. $h = n - 1$

ומהו הגובה הממוצע ?

גובה ממוצע

ברור שצורת העץ נקבעת על פי סדר ההכנסה (למשל הסדר 1,2,3יוצר שרשרת לעומת 2,1,3 שיוצר עץ מאוזן).

מספר אפשרויות (הסדרים) להכניס n צמתים לעץ הוא n!.

נסמן ב- h(i) את גובה העץ הנוצר בסדר ה-i.

הגובה הממוצע מוגדר כדלקמן:

$$\bar{h} = \frac{1}{n!}\sum_{i=1}^{n!} h(i)$$

גובה ממוצע

ברור שצורת העץ נקבעת על פי סדר ההכנסה (למשל הסדר 1,2,3 יוצר שרשרת לעומת 2,1,3 שיוצר עץ מאוזן).

מספר אפשרויות (הסדרים) להכניס n צמתים לעץ הוא n!.

נסמן ב- h(i) את גובה העץ הנוצר בסדר ה-i.

הגובה הממוצע מוגדר כדלקמן:

$$\bar{h} = \frac{1}{n!}\sum_{i=1}^{n!} h(i)$$

ניתן להראות שהגובה הממוצע שייך לקבוצה- O(log n) כלומר בממוצע כל הפעולות מתבצעות בזמן O(log n).

גובה ממוצע

ברור שצורת העץ נקבעת על פי סדר ההכנסה (למשל הסדר 1,2,3יוצר שרשרת לעומת 2,1,3 שיוצר עץ מאוזן).

מספר אפשרויות (הסדרים) להכניס n צמתים לעץ הוא !n.

נסמן ב- h(i) את גובה העץ הנוצר בסדר ה-i.

הגובה הממוצע מוגדר כדלקמן:

$$\bar{h} = \frac{1}{n!}\sum_{i=1}^{n!} h(i)$$

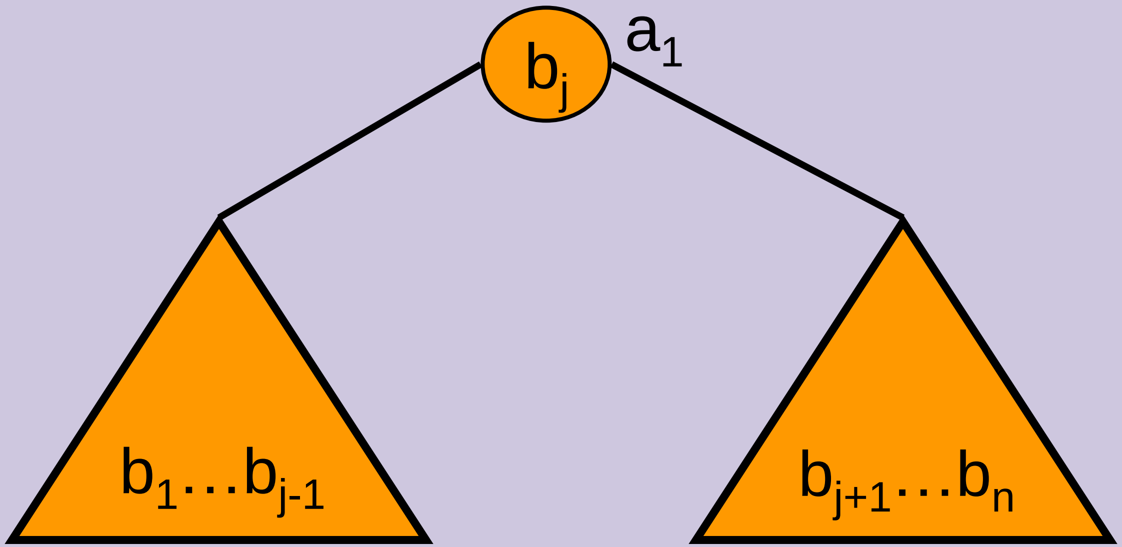
ניתן להראות שהגובה הממוצע שייך לקבוצה- O(log n) כלומר בממוצע כל הפעולות מתבצעות בזמן O(log n).

ההוכחה (עמודים 254-258 בספר הלימוד) מושמטת. נבחן טענה דומה אך קלה יותר להוכחה: זמן בניה ממוצע של עץ חיפוש בינרי הוא O(n log n).

זמן בניה צפוי של עץ חיפוש בינרי

נחשב את זמן בנית עץ אקראי המתקבל מהכנסת פרמוטציה אקראית $a_1 \ldots a_n$ לעץ ריק.

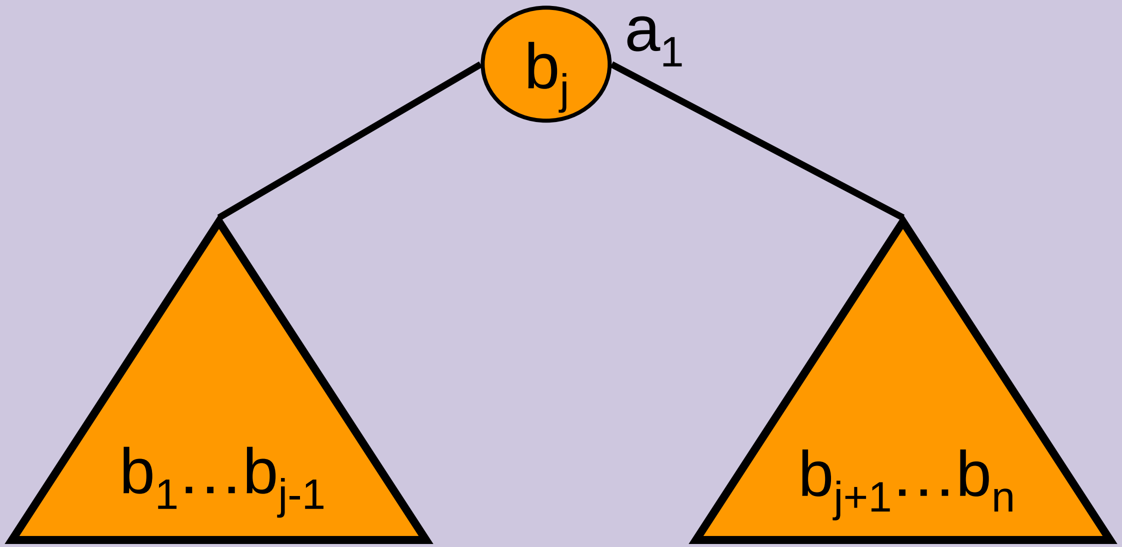
נניח שסדר האיברים הוא $b_1 \ldots b_n$



זמן בניה צפוי של עץ חיפוש בינרי

נחשב את זמן בנית עץ אקראי המתקבל מהכנסת פרמוטציה אקראית $a_1 \ldots a_n$ לעץ ריק.

נניח שסדר האיברים הוא $b_1 \ldots b_n$



נסמן ב- $T(n)$ את מספר ההשוואות הממוצע הדרוש לבניית עץ בן n צמתים.

לפיכך $T(j-1)$ הוא מספר ההשוואות הממוצע הדרוש לבניית עץ בן j-1 צמתים

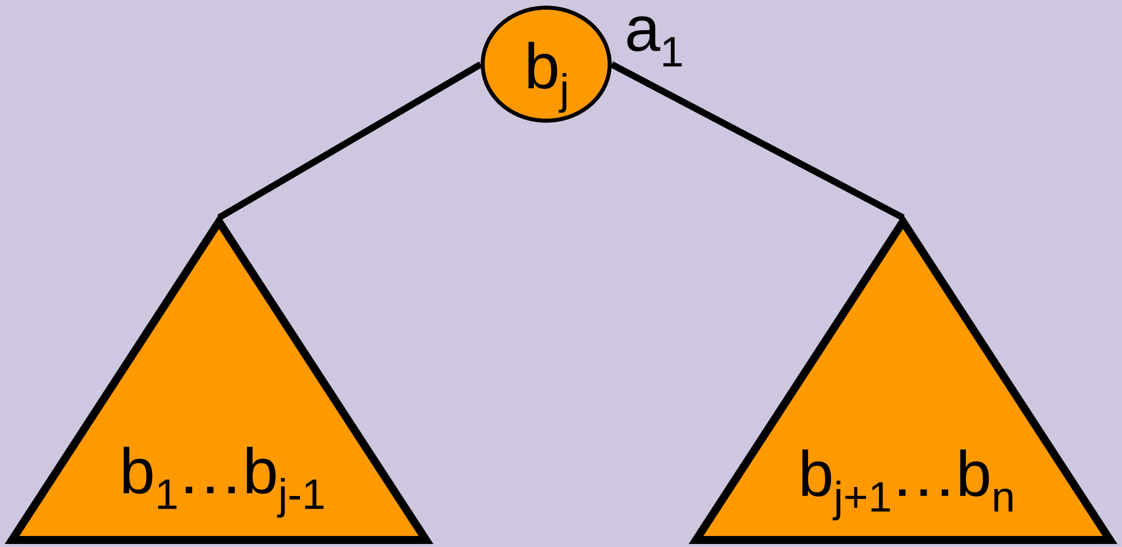
ו- $T(n-j)$ הוא מספר ההשוואות הממוצע הדרוש לבניית עץ בן n-j צמתים.

משוואת ההפרשים המתאימה:

זמן בניה צפוי של עץ חיפוש בינרי

נחשב את זמן בנית עץ אקראי המתקבל מהכנסת פרמוטציה אקראית $a_1 \ldots a_n$ לעץ ריק.

נניח שסדר האיברים הוא $b_1 \ldots b_n$



נסמן ב- T(n) את מספר ההשוואות הממוצע הדרוש לבניית עץ בן n צמתים.

לפיכך T(j-1) הוא מספר ההשוואות הממוצע הדרוש לבניית עץ בן j-1 צמתים

ו- T(n-j) הוא מספר ההשוואות הממוצע הדרוש לבניית עץ בן n-j צמתים.

משוואת ההפרשים המתאימה:

$$T(n) = \frac{1}{n}\sum_{j=1}^{n}\left[(n-1) + T(j-1) + T(n-j)\right] \qquad (T(0)=0)$$

פתרון משוואת ההפרשים

$$T(n)=\frac{1}{n}\sum_{j=1}^{n}\left[(n-1)+T(j-1)+T(n-j)\right]=(n-1)+\frac{2}{n}\sum_{j=0}^{n-1}T(j)$$

פתרון משוואת ההפרשים

$$T(n) = \frac{1}{n}\sum_{j=1}^{n}\left[(n-1) + T(j-1) + T(n-j)\right] = (n-1) + \frac{2}{n}\sum_{j=0}^{n-1}T(j)$$

$$nT(n) = n(n-1) + 2\sum_{j=0}^{n-1}T(j)$$

פתרון משוואת ההפרשים

$$T(n) = \frac{1}{n}\sum_{j=1}^{n}\left[(n-1) + T(j-1) + T(n-j)\right] = (n-1) + \frac{2}{n}\sum_{j=0}^{n-1}T(j)$$

$$nT(n) = n(n-1) + 2\sum_{j=0}^{n-1}T(j)$$

$$-$$

$$(n-1)T(n-1)) = (n-1)(n-2) + 2\sum_{j=0}^{n-2}T(j)$$

פתרון משוואת ההפרשים

$$T(n)=\frac{1}{n}\sum_{j=1}^{n}\left[(n-1)+T(j-1)+T(n-j)\right]=(n-1)+\frac{2}{n}\sum_{j=0}^{n-1}T(j)$$

$$nT(n)=n(n-1)+2\sum_{j=0}^{n-1}T(j)$$

$-$

$$(n-1)T(n-1))=(n-1)(n-2)+2\sum_{j=0}^{n-2}T(j)$$

$$nT(n)-(n-1)T(n-1)=2(n-1)+2T(n-1)$$

פתרון משוואת ההפרשים

$$T(n)=\frac{1}{n}\sum_{j=1}^{n}\left[(n-1)+T(j-1)+T(n-j)\right]=(n-1)+\frac{2}{n}\sum_{j=0}^{n-1}T(j)$$

$$nT(n)=n(n-1)+2\sum_{j=0}^{n-1}T(j)$$

$$-$$

$$(n-1)T(n-1))=(n-1)(n-2)+2\sum_{j=0}^{n-2}T(j)$$

$$nT(n)-(n-1)T(n-1)=2(n-1)+2T(n-1)$$

$$\boxed{nT(n)=2n-2+(n+1)T(n-1)}$$

פתרון משוואת ההפרשים

$$nT(n) = 2n - 2 + (n+1)T(n-1)$$

פתרון משוואת ההפרשים

$$nT(n) = 2n - 2 + (n+1)T(n-1)$$

$$\frac{T(n)}{n+1} = \frac{2}{n+1} - \frac{2}{n(n+1)} + \frac{T(n-1)}{n}$$

פתרון משוואת ההפרשים

$$nT(n) = 2n - 2 + (n+1)T(n-1)$$

$$\frac{T(n)}{n+1} = \frac{2}{n+1} - \frac{2}{n(n+1)} + \frac{T(n-1)}{n} < \frac{2}{n+1} + \frac{T(n-1)}{n}$$

פתרון משוואת ההפרשים

$$nT(n) = 2n - 2 + (n+1)T(n-1)$$

$$\frac{T(n)}{n+1} = \frac{2}{n+1} - \frac{2}{n(n+1)} + \frac{T(n-1)}{n} < \frac{2}{n+1} + \frac{T(n-1)}{n}$$

$$\frac{T(n)}{n+1} < 2\sum_{j=1}^{n} \frac{1}{j+1} + \frac{T(0)}{1} = 2(H_{n+1} - 1) + 0 \leq 2H_{n+1}$$

פתרון משוואת ההפרשים

$$nT(n) = 2n - 2 + (n+1)T(n-1)$$

$$\frac{T(n)}{n+1} = \frac{2}{n+1} - \frac{2}{n(n+1)} + \frac{T(n-1)}{n} < \frac{2}{n+1} + \frac{T(n-1)}{n}$$

$$\frac{T(n)}{n+1} < 2\sum_{j=1}^{n} \frac{1}{j+1} + \frac{T(0)}{1} = 2(H_{n+1} - 1) + 0 \le 2H_{n+1}$$

כאשר H_n הוא הטור ההרמוני:

$$H_n = \sum_{i=1}^{n} \frac{1}{i} = \ln n + 0.57721... + o(\frac{1}{n}) = O(\ln n)$$

קבוע אוילר

פתרון משוואת ההפרשים

$$nT(n) = 2n - 2 + (n+1)T(n-1)$$

$$\frac{T(n)}{n+1} = \frac{2}{n+1} - \frac{2}{n(n+1)} + \frac{T(n-1)}{n} < \frac{2}{n+1} + \frac{T(n-1)}{n}$$

$$\frac{T(n)}{n+1} < 2\sum_{j=1}^{n} \frac{1}{j+1} + \frac{T(0)}{1} = 2(H_{n+1} - 1) + 0 \le 2H_{n+1}$$

כאשר H_n הוא הטור ההרמוני:

$$H_n = \sum_{i=1}^{n} \frac{1}{i} = \ln n + 0.57721... + o(\frac{1}{n}) = O(\ln n)$$

קבוע אוילר

$$\frac{T(n)}{n+1} \le 2H_{n+1} = 2\ln(n+1) + O(1)$$

פתרון משוואת ההפרשים

$$nT(n) = 2n - 2 + (n+1)T(n-1)$$

$$\frac{T(n)}{n+1} = \frac{2}{n+1} - \frac{2}{n(n+1)} + \frac{T(n-1)}{n} < \frac{2}{n+1} + \frac{T(n-1)}{n}$$

$$\frac{T(n)}{n+1} < 2\sum_{j=1}^{n} \frac{1}{j+1} + \frac{T(0)}{1} = 2(H_{n+1} - 1) + 0 \le 2H_{n+1}$$

כאשר H_n הוא הטור ההרמוני:

$$H_n = \sum_{i=1}^{n} \frac{1}{i} = \ln n + 0.57721\ldots + o(\frac{1}{n}) = O(\ln n)$$

קבוע אוילר

$$\frac{T(n)}{n+1} \le 2H_{n+1} = 2\ln(n+1) + O(1)$$

$$T(n) \le 2(n+1)\ln(n+1) + O(n+1) = 2n\ln n + O(n)$$

trees