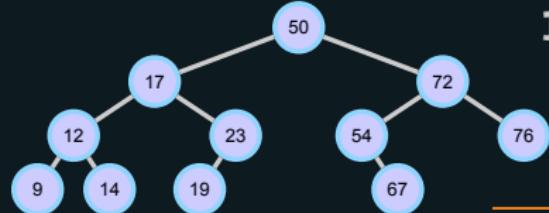


יסודות האלגוריתמים והסיבוכיות

תרגול 04 - עצי חיפוש בינאריים





למדנו על כמה מבני נתונים המאפשרים מספר פעולות:

| מבנה | רישמה מקוורת | מערך ממויין |
|------------|-------------------------|---|
| בנייה | $\mathcal{O}(n \log n)$ | $\mathcal{O}(n)$ |
| הוספת איבר | $\mathcal{O}(n)$ | $\mathcal{O}(1)$ |
| הוצאת איבר | $\mathcal{O}(n)$ | $\mathcal{O}(1) \text{ או } \mathcal{O}(n)$ |
| חיפוש | $\mathcal{O}(\log n)$ | $\mathcal{O}(n)$ |



למדנו על כמה מבני נתונים המאפשרים מספר פעולות:

| | מבנה | רישימה מקוורת | עץ חיפוש בינהארית | מערך ממויין |
|------------|------|---------------|-------------------|-------------|
| הוספת איבר | | | | בנייה |
| הוצאת איבר | | | | הוספת איבר |
| חיפוש | | | | חיפוש |



למדנו על כמה מבני נתונים המאפשרים מספר פעולות:

| מבנה | רשימה מקוורת | עץ חיפוש בינארית | מערך ממויין |
|------------|---------------------------------------|--------------------------------------|------------------|
| בנייה | $\mathcal{O}(n \times \text{height})$ | $\mathcal{O}(n)$ | $n \log n$ |
| הוספת איבר | $\mathcal{O}(\text{height})$ | $\mathcal{O}(1)$ | $\mathcal{O}(n)$ |
| הוצאת איבר | $\mathcal{O}(\text{height})$ | $\mathcal{O}(1)$ או $\mathcal{O}(n)$ | $\mathcal{O}(n)$ |
| חיפוש | $\mathcal{O}(\text{height})$ | $\mathcal{O}(n)$ | $n \log n$ |

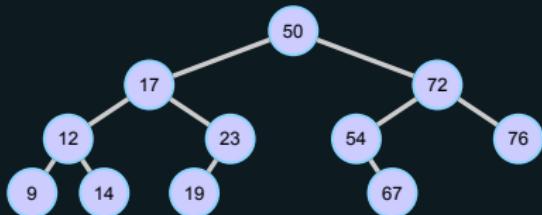
עץ חיפוש ביןארית הוא גրף הבנוי כך שלכל צומת יש עד שני ילדים, ימני ושמאלי. כמו כן לכל צומת יש צומת אב אחד למעט השורש.

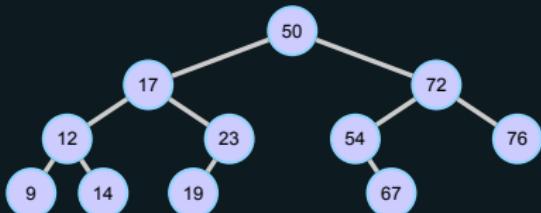
למדנו על כמה מבני נתונים המאפשרים מספר פעולות:

| | מערך ממויין | רשימה מקוורת | עץ חיפוש בינארי |
|------------|---------------------------------------|--------------------------------------|------------------|
| בנייה | $\mathcal{O}(n \times \text{height})$ | $\mathcal{O}(n)$ | $n \log n$ |
| הוספת איבר | $\mathcal{O}(\text{height})$ | $\mathcal{O}(1)$ | $\mathcal{O}(n)$ |
| הוצאת איבר | $\mathcal{O}(\text{height})$ | $\mathcal{O}(1)$ או $\mathcal{O}(n)$ | $\mathcal{O}(n)$ |
| חיפוש | $\mathcal{O}(\text{height})$ | $\mathcal{O}(n)$ | $n \log n$ |

עץ חיפוש בינארי הוא גրף הבנוי כך שלכל צומת יש עד שני ילדים, ימני ושמאלי. כמו כן לכל צומת יש צומת אב אחד למעט השורש.

עבור כל צומת u בעץ חיפוש בינארי מתקיים: המפתח של u קטן מהמפתחות של כל הצמתים בתת העץ הימני שלו וגדול מכל המפתחות בתת העץ השמאלי.





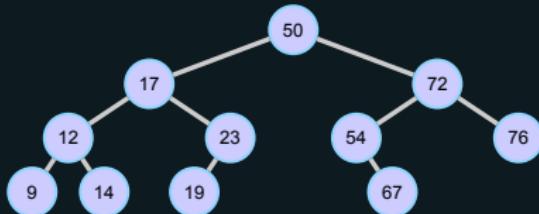
העץ מימין - עץ ימין ⇐ שורש ⇐ עץ שמאל - In Order

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
```

```
Print x.key
```

```
if x.right ≠ Ø then InOrder(x.right)
```



תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ ימין

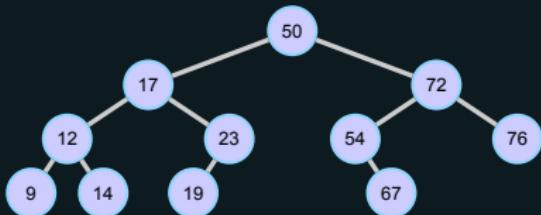
InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

9



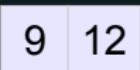
תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ ימין

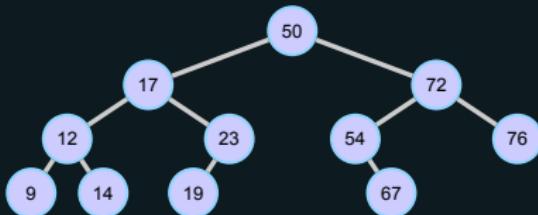
InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)





In Order - תחת עץ שמאל \Leftarrow שורש \Leftarrow תחת עץ ימין

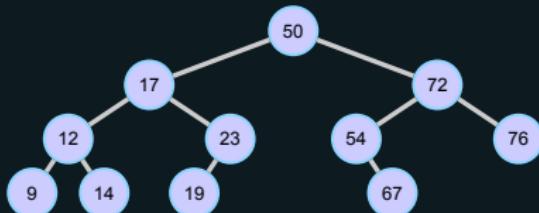
InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)





תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ ימין

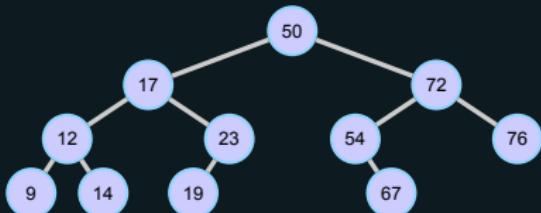
InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | |
|---|----|----|----|
| 9 | 12 | 14 | 17 |
|---|----|----|----|



תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ ימין

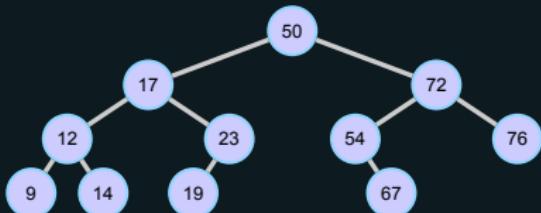
InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | |
|---|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 |
|---|----|----|----|----|



תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ ימין

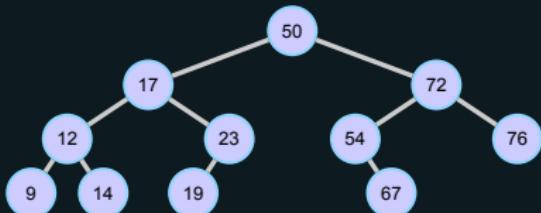
InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | |
|---|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 |
|---|----|----|----|----|----|



תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ שמאל - In Order

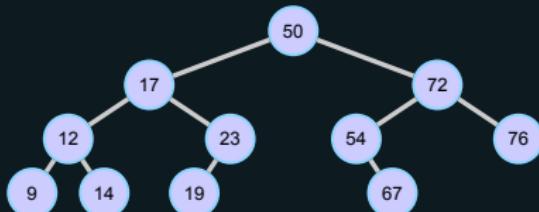
InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | |
|---|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 |
|---|----|----|----|----|----|----|



In Order - תחת עץ שמאל \Leftarrow שורש \Leftarrow תחת עץ ימין

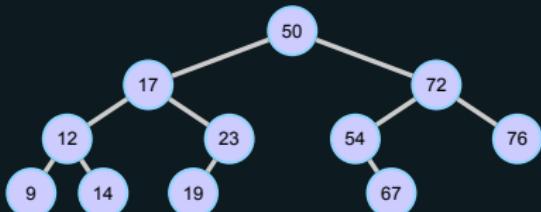
InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | |
|---|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 |
|---|----|----|----|----|----|----|----|



תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ ימין

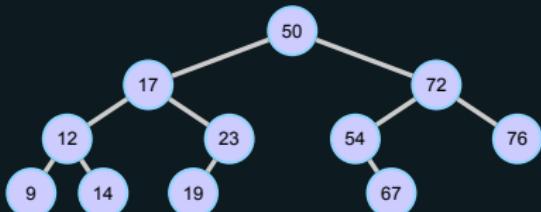
InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | |
|---|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 |
|---|----|----|----|----|----|----|----|----|



תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ ימין

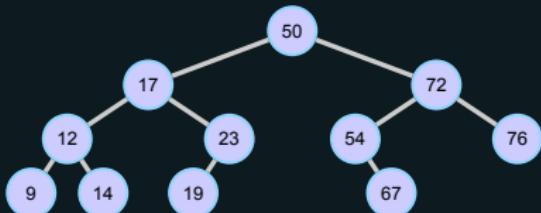
InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 |
|---|----|----|----|----|----|----|----|----|----|



תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ ימין

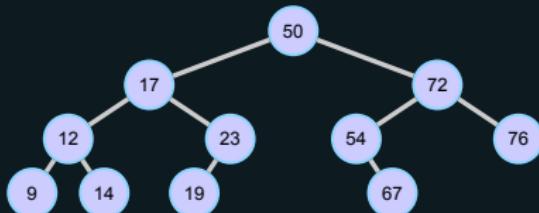
InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|



הנץ עץ שמאל \Leftarrow שורש \Leftarrow הנץ עץ ימין

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

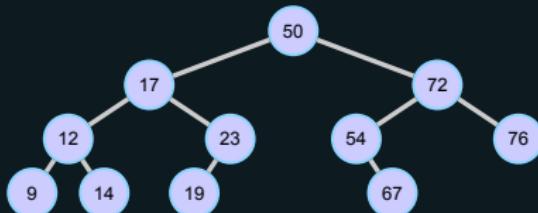
Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

הנץ עץ ימין \Leftarrow שורש \Leftarrow הנץ עץ שמאל

(Out Order)



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין

InOrder(x):

```
if x.left  $\neq \emptyset$  then InOrder(x.left)
```

Print x.key

```
if x.right  $\neq \emptyset$  then InOrder(x.right)
```

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order ())

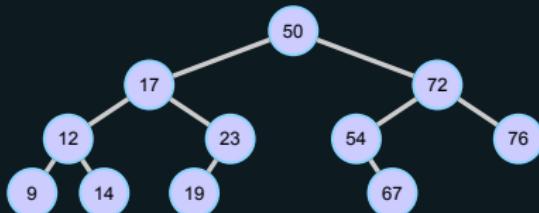
שורש \Leftarrow תת עץ שמאל \Leftarrow Pre Order

PreOrder(x):

Print x.key

```
if x.left  $\neq \emptyset$  then PreOrder(x.left)
```

```
if x.right  $\neq \emptyset$  then PreOrder(x.right)
```



העץ מתחת לשורש יימין - תחילה שורש יימין - תחילה שורש יימין - In Order

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
```

Print x.key

```
if x.right ≠ Ø then InOrder(x.right)
```

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

העץ מתחת לשורש יימין - Out Order

(In Order ()) הפוך מ-Order

שורש יימין - תחילה שמאלי - Pre Order

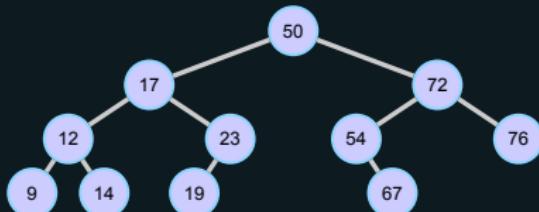
PreOrder(x):

Print x.key

```
if x.left ≠ Ø then PreOrder(x.left)
```

```
if x.right ≠ Ø then PreOrder(x.right)
```

| |
|----|
| 50 |
|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין - In Order

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין - Out Order \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order ()) הפוך מ-Order

שורש \Leftarrow תת עץ שמאל \Leftarrow Pre Order

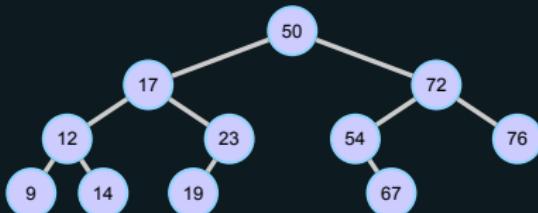
PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | |
|----|----|
| 50 | 17 |
|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין - In Order

InOrder(x):

```
if x.left  $\neq \emptyset$  then InOrder(x.left)
```

Print x.key

```
if x.right  $\neq \emptyset$  then InOrder(x.right)
```

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין - Out Order \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order) הפק מ-Order

שורש \Leftarrow תת עץ שמאל \Leftarrow Pre Order

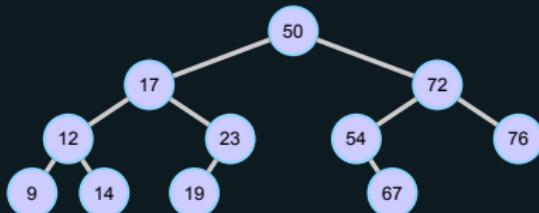
PreOrder(x):

Print x.key

```
if x.left  $\neq \emptyset$  then PreOrder(x.left)
```

```
if x.right  $\neq \emptyset$  then PreOrder(x.right)
```

| | | |
|----|----|----|
| 50 | 17 | 12 |
|----|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין - In Order

InOrder(x):

```
if x.left  $\neq \emptyset$  then InOrder(x.left)
```

Print x.key

```
if x.right  $\neq \emptyset$  then InOrder(x.right)
```

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין - Out Order \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order)

שורש \Leftarrow תת עץ שמאל \Leftarrow Pre Order

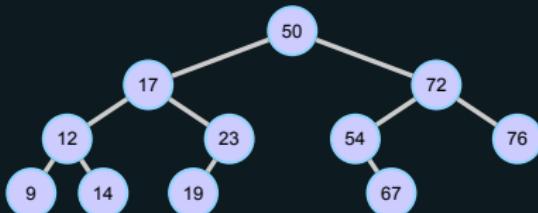
PreOrder(x):

Print x.key

```
if x.left  $\neq \emptyset$  then PreOrder(x.left)
```

```
if x.right  $\neq \emptyset$  then PreOrder(x.right)
```

| | | | |
|----|----|----|---|
| 50 | 17 | 12 | 9 |
|----|----|----|---|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין

InOrder(x):

```
if x.left  $\neq \emptyset$  then InOrder(x.left)
```

Print x.key

```
if x.right  $\neq \emptyset$  then InOrder(x.right)
```

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order)

שורש \Leftarrow תת עץ שמאל \Leftarrow תת עץ ימין

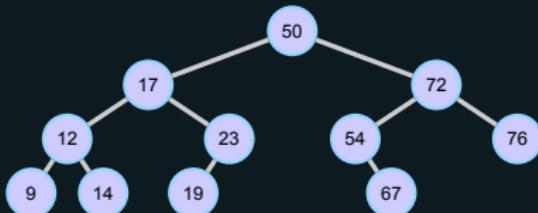
PreOrder(x):

Print x.key

```
if x.left  $\neq \emptyset$  then PreOrder(x.left)
```

```
if x.right  $\neq \emptyset$  then PreOrder(x.right)
```

| | | | | |
|----|----|----|---|----|
| 50 | 17 | 12 | 9 | 14 |
|----|----|----|---|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין

InOrder(x):

```
if x.left  $\neq \emptyset$  then InOrder(x.left)
```

Print x.key

```
if x.right  $\neq \emptyset$  then InOrder(x.right)
```

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order)

שורש \Leftarrow תת עץ שמאל \Leftarrow תת עץ ימין

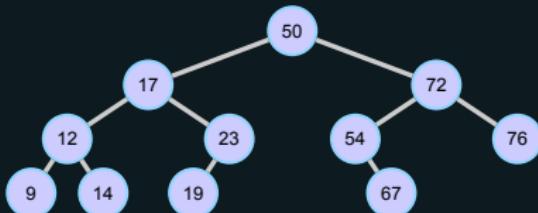
PreOrder(x):

Print x.key

```
if x.left  $\neq \emptyset$  then PreOrder(x.left)
```

```
if x.right  $\neq \emptyset$  then PreOrder(x.right)
```

| | | | | | |
|----|----|----|---|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 |
|----|----|----|---|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין

InOrder(x):

```
if x.left  $\neq \emptyset$  then InOrder(x.left)
```

Print x.key

```
if x.right  $\neq \emptyset$  then InOrder(x.right)
```

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order)

שורש \Leftarrow תת עץ שמאל \Leftarrow תת עץ ימין

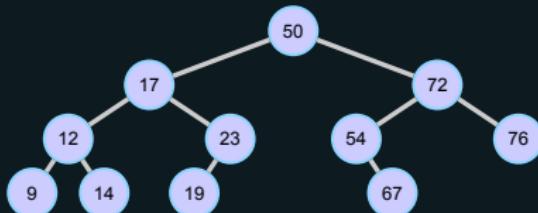
PreOrder(x):

Print x.key

```
if x.left  $\neq \emptyset$  then PreOrder(x.left)
```

```
if x.right  $\neq \emptyset$  then PreOrder(x.right)
```

| | | | | | | |
|----|----|----|---|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 |
|----|----|----|---|----|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין

InOrder(x):

```
if x.left  $\neq \emptyset$  then InOrder(x.left)
```

Print x.key

```
if x.right  $\neq \emptyset$  then InOrder(x.right)
```

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order)

שורש \Leftarrow תת עץ שמאל \Leftarrow תת עץ ימין

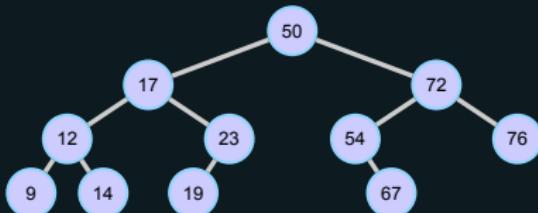
PreOrder(x):

Print x.key

```
if x.left  $\neq \emptyset$  then PreOrder(x.left)
```

```
if x.right  $\neq \emptyset$  then PreOrder(x.right)
```

| | | | | | | | |
|----|----|----|---|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 |
|----|----|----|---|----|----|----|----|



תת עץ ימין ← שורש ← תת עץ ימין - In Order

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין ← שורש ← תת עץ שמאל - Out Order

(In Order ()) הפוך מ-Order

תת עץ שמאל ← שורש ← תת עץ ימין - Pre Order

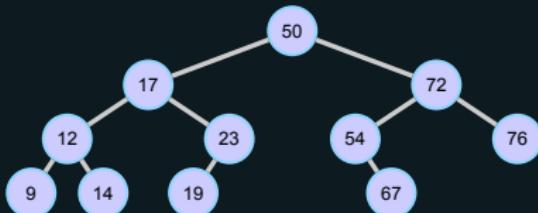
PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | | | | | | | | |
|----|----|----|---|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 |
|----|----|----|---|----|----|----|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין

InOrder(x):

```
if x.left  $\neq \emptyset$  then InOrder(x.left)
```

Print x.key

```
if x.right  $\neq \emptyset$  then InOrder(x.right)
```

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order)

שורש \Leftarrow תת עץ שמאל \Leftarrow תת עץ ימין

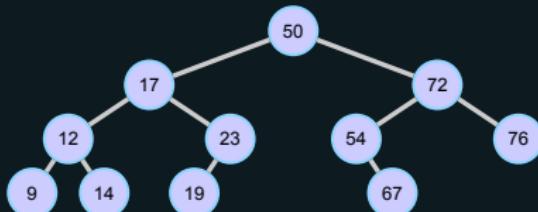
PreOrder(x):

Print x.key

```
if x.left  $\neq \emptyset$  then PreOrder(x.left)
```

```
if x.right  $\neq \emptyset$  then PreOrder(x.right)
```

| | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 |
|----|----|----|---|----|----|----|----|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין

InOrder(x):

```
if x.left  $\neq \emptyset$  then InOrder(x.left)
```

Print x.key

```
if x.right  $\neq \emptyset$  then InOrder(x.right)
```

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order)

שורש \Leftarrow תת עץ שמאל \Leftarrow תת עץ ימין

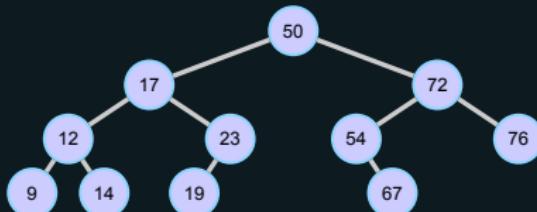
PreOrder(x):

Print x.key

```
if x.left  $\neq \emptyset$  then PreOrder(x.left)
```

```
if x.right  $\neq \emptyset$  then PreOrder(x.right)
```

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין - In Order

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין - Out Order \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order ()) הפוך מ-Order

שורש \Leftarrow תת עץ שמאל \Leftarrow Pre Order

PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|

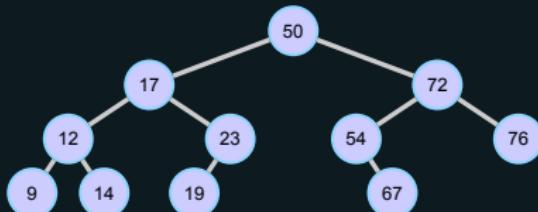
תת עץ שמאל \Leftarrow תת עץ ימין \Leftarrow שורש - Post Order

PostOrder(x):

if $x.\text{left} \neq \emptyset$ then PostOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PostOrder($x.\text{right}$)

Print $x.\text{key}$



הנתונים נראים כך: In Order - תחת עץ שמאל => שורש => תחת עץ ימין

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

הנתונים נראים כך: Out Order - תחת עץ ימין => שורש => תחת עץ שמאל

(In Order)

הנתונים נראים כך: Pre Order - שורש => תחת עץ שמאל => תחת עץ ימין

PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|

הנתונים נראים כך: Post Order - תחת עץ שמאל => תחת עץ ימין => שורש

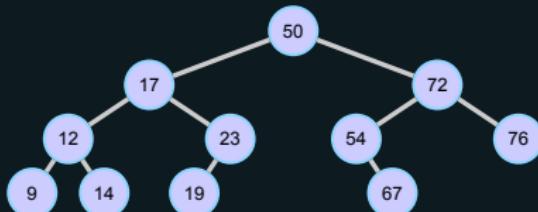
PostOrder(x):

if $x.\text{left} \neq \emptyset$ then PostOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PostOrder($x.\text{right}$)

Print $x.\text{key}$

| |
|---|
| 9 |
|---|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין - In Order

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין - Out Order \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order () הפוך מ-)

שורש \Leftarrow תת עץ שמאל \Leftarrow Pre Order

PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|

תת עץ שמאל \Leftarrow תת עץ ימין \Leftarrow שורש - Post Order

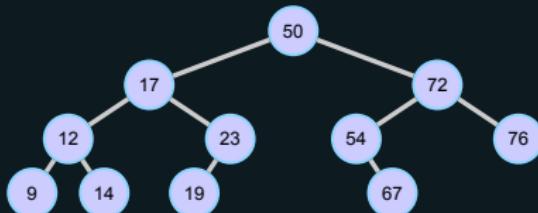
PostOrder(x):

if $x.\text{left} \neq \emptyset$ then PostOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PostOrder($x.\text{right}$)

Print $x.\text{key}$

| | |
|---|----|
| 9 | 14 |
|---|----|



הנחות - עץ שמאלי => שורש => עץ ימני

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

הנחות - עץ ימני => שורש => עץ שמאלי

(In Order)

הנחות - שורש => עץ שמאלי => Pre Order

PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|

הנחות - עץ שמאלי => עץ ימני => שורש

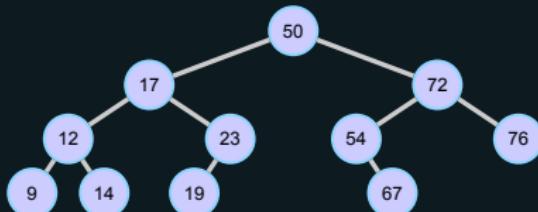
PostOrder(x):

if $x.\text{left} \neq \emptyset$ then PostOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PostOrder($x.\text{right}$)

Print $x.\text{key}$

| | | |
|---|----|----|
| 9 | 14 | 12 |
|---|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין - In Order

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין - Out Order \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order () הפוך מ-)

שורש \Leftarrow תת עץ שמאל \Leftarrow Pre Order

PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|

תת עץ שמאל \Leftarrow תת עץ ימין \Leftarrow שורש - Post Order

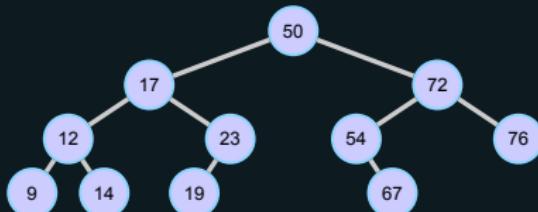
PostOrder(x):

if $x.\text{left} \neq \emptyset$ then PostOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PostOrder($x.\text{right}$)

Print $x.\text{key}$

| | | | |
|---|----|----|----|
| 9 | 14 | 12 | 19 |
|---|----|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין - In Order

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין - Out Order \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order)

שורש \Leftarrow תת עץ שמאל \Leftarrow Pre Order

PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|

תת עץ שמאל \Leftarrow תת עץ ימין \Leftarrow שורש - Post Order

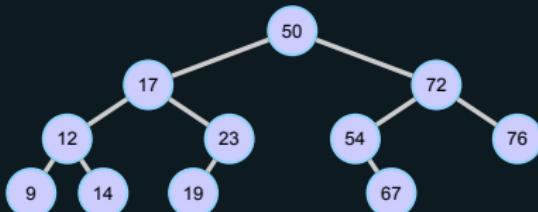
PostOrder(x):

if $x.\text{left} \neq \emptyset$ then PostOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PostOrder($x.\text{right}$)

Print $x.\text{key}$

| | | | | |
|---|----|----|----|----|
| 9 | 14 | 12 | 19 | 23 |
|---|----|----|----|----|



- תחת עץ שמאל \Leftarrow שורש \Leftarrow תחת עץ ימין

InOrder(x):

if $x.\text{left} \neq \emptyset$ **then** InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ **then** InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

- תחת עץ ימין \Leftarrow שורש \Leftarrow תחת עץ שמאל

(In Order)

- שורש \Leftarrow תחת עץ שמאל \Leftarrow Pre Order

PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ **then** PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ **then** PreOrder($x.\text{right}$)

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|

- תחת עץ שמאל \Leftarrow תחת עץ ימין \Leftarrow שורש

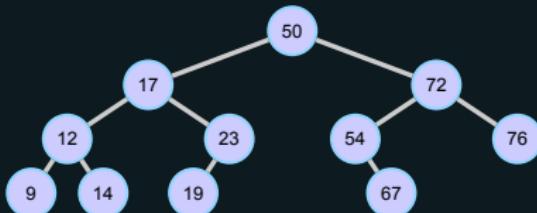
PostOrder(x):

if $x.\text{left} \neq \emptyset$ **then** PostOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ **then** PostOrder($x.\text{right}$)

Print $x.\text{key}$

| | | | | | |
|---|----|----|----|----|----|
| 9 | 14 | 12 | 19 | 23 | 17 |
|---|----|----|----|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין - In Order

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין - Out Order \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order) הפק מ-Order

שורש \Leftarrow תת עץ שמאל \Leftarrow Pre Order

PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|

תת עץ שמאל \Leftarrow תת עץ ימין \Leftarrow שורש - Post Order

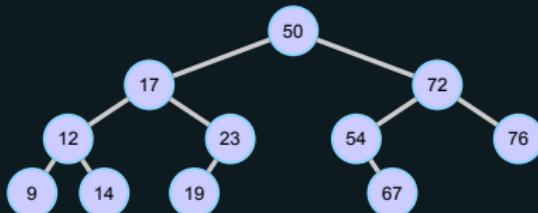
PostOrder(x):

if $x.\text{left} \neq \emptyset$ then PostOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PostOrder($x.\text{right}$)

Print $x.\text{key}$

| | | | | | | |
|---|----|----|----|----|----|----|
| 9 | 14 | 12 | 19 | 23 | 17 | 67 |
|---|----|----|----|----|----|----|



- תחת עץ שמאל \Leftarrow שורש \Leftarrow תחת עץ ימין

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

- תחת עץ ימין \Leftarrow שורש \Leftarrow תחת עץ שמאל

(In Order)

- שורש \Leftarrow תחת עץ שמאל \Leftarrow Pre Order

PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|

- תחת עץ ימין \Leftarrow תחת עץ שמאל \Leftarrow שורש

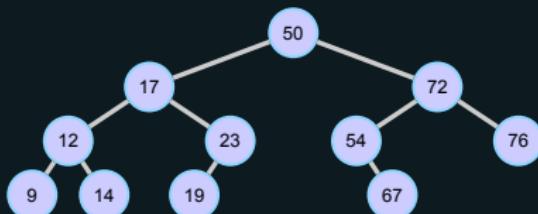
PostOrder(x):

if $x.\text{left} \neq \emptyset$ then PostOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PostOrder($x.\text{right}$)

Print $x.\text{key}$

| | | | | | | | |
|---|----|----|----|----|----|----|----|
| 9 | 14 | 12 | 19 | 23 | 17 | 67 | 54 |
|---|----|----|----|----|----|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין - In Order

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין - Out Order \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order) הפק מ-Order

שורש \Leftarrow תת עץ שמאל \Leftarrow Pre Order

PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|

תת עץ שמאל \Leftarrow תת עץ ימין \Leftarrow שורש - Post Order

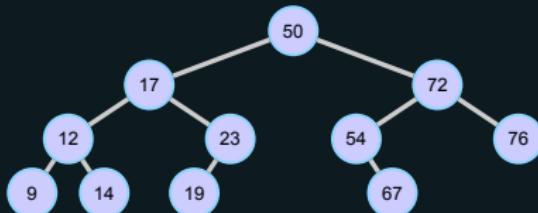
PostOrder(x):

if $x.\text{left} \neq \emptyset$ then PostOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PostOrder($x.\text{right}$)

Print $x.\text{key}$

| | | | | | | | | |
|---|----|----|----|----|----|----|----|----|
| 9 | 14 | 12 | 19 | 23 | 17 | 67 | 54 | 76 |
|---|----|----|----|----|----|----|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין - In Order

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין - Out Order \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order () הפוך מ-)

שורש \Leftarrow תת עץ שמאל \Leftarrow Pre Order

PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|

תת עץ שמאל \Leftarrow תת עץ ימין \Leftarrow שורש - Post Order

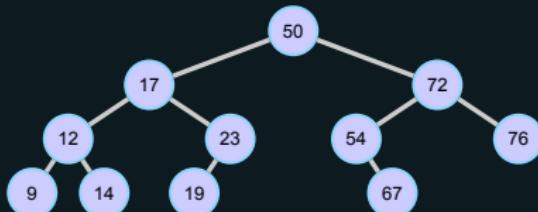
PostOrder(x):

if $x.\text{left} \neq \emptyset$ then PostOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PostOrder($x.\text{right}$)

Print $x.\text{key}$

| | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|
| 9 | 14 | 12 | 19 | 23 | 17 | 67 | 54 | 76 | 72 |
|---|----|----|----|----|----|----|----|----|----|



תת עץ שמאל \Leftarrow שורש \Leftarrow תת עץ ימין - In Order

InOrder(x):

if $x.\text{left} \neq \emptyset$ then InOrder($x.\text{left}$)

Print $x.\text{key}$

if $x.\text{right} \neq \emptyset$ then InOrder($x.\text{right}$)

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 12 | 14 | 17 | 19 | 23 | 50 | 54 | 67 | 72 | 76 |
|---|----|----|----|----|----|----|----|----|----|----|

תת עץ ימין - Out Order \Leftarrow שורש \Leftarrow תת עץ שמאל

(In Order) הפק מ-Order

שורש \Leftarrow תת עץ שמאל \Leftarrow Pre Order

PreOrder(x):

Print $x.\text{key}$

if $x.\text{left} \neq \emptyset$ then PreOrder($x.\text{left}$)

if $x.\text{right} \neq \emptyset$ then PreOrder($x.\text{right}$)

| | | | | | | | | | | |
|----|----|----|---|----|----|----|----|----|----|----|
| 50 | 17 | 12 | 9 | 14 | 23 | 19 | 72 | 54 | 67 | 76 |
|----|----|----|---|----|----|----|----|----|----|----|

תת עץ שמאל \Leftarrow תת עץ ימין \Leftarrow שורש - Post Order

PostOrder(x):

if $x.\text{left} \neq \emptyset$ then PostOrder($x.\text{left}$)

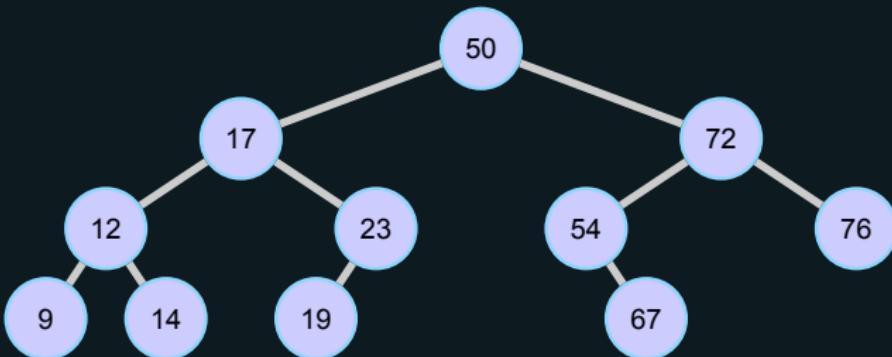
if $x.\text{right} \neq \emptyset$ then PostOrder($x.\text{right}$)

Print $x.\text{key}$

| | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|
| 9 | 14 | 12 | 19 | 23 | 17 | 67 | 54 | 76 | 72 | 50 |
|---|----|----|----|----|----|----|----|----|----|----|

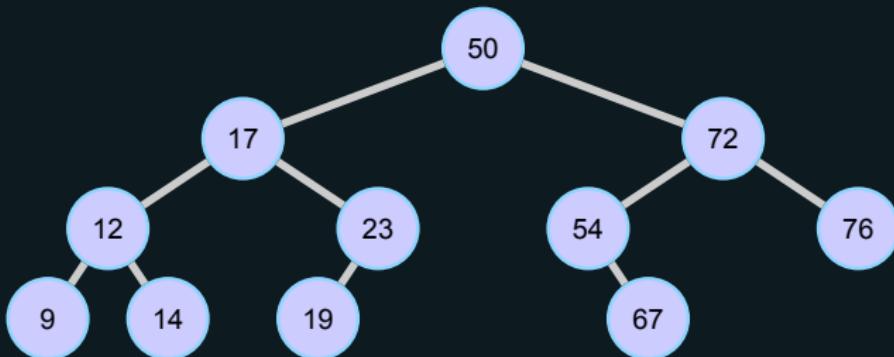


כתבו תוכנית המחזירה מפתח מינימאלי בעץ חיפוש ביארי.





כתבו תוכנית המחזיר מפתח מינימאלי בעץ חיפוש בינארי.



$\text{Min}(x)$:

```

while  $x.\text{left} \neq \emptyset$  do
   $x = x.\text{left}$ 
return  $x.\text{key}$ 
  
```

על מנת למצוא את המפתח המינימלי
צריך למצוא את האיבר הכי שמאלי,
נפעיל את Min עם שורש העץ כקלט.



PreOrder(x):

```

Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
    
```

InOrder(x):

```

if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
    
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

.*a, b, c, d, e, g, h, j, f* :pre-order
.c, d, b, a, h, g, j, e, f :in-order
רמז: העזרו ברקורייה.



PreOrder(x):

```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

. **a**, b, c, d, e, g, h, j, f :pre-order
. c, d, b, a, h, g, j, e, f :in-order
רמז: העזרו ברקורייה.

- נשים לב ששורש העץ מופיע בתחילת סריקת pre-order.



PreOrder(x):

```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסויים. שחזור את העץ המקורי.

. **a, b, c, d, e, g, h, j, f** :pre-order
. **c,d,b, a, h,g,j,e,f** :in-order
רמז: העזרו ברקוריוסיה.

po=[**a, b, c, d, e, g, h, j, f**]
io=[**c, d, b, a, h, g, j, e, f**]

- נשים לב ששורש העץ מופיע בתחילת pre-order.
- "עפ" סריקת order-in כל האיברים ממשימים לשורש חייבים להיות גם לימינו בעץ.



PreOrder(x):

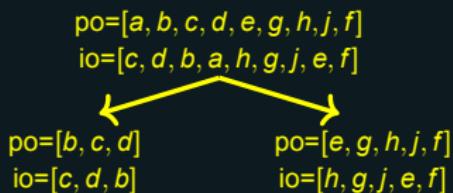
```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסויים. שחזרו את העץ המקורי.

- . **a, b, c, d, e, g, h, j, f** :pre-order
 - . **c,d,b, a, h,g,j,e,f** :in-order
- רמז: העזרו ברקוריוסיה.



- נשים לב ששורש העץ מופיע בתחילת pre-order.
- סריקת order-in כל האיברים שמופיעים לשורש חייבים להיות גם לימינו בעץ.
- כתע נוכל לחזור על הפעולה על תתי העץ עד שנשחזר את כל העץ.



PreOrder(x):

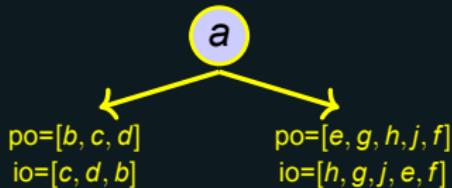
```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

.*a, b, c, d, e, g, h, j, f* :pre-order
.*c, d, b, a, h, g, j, e, f* :in-order
רמז: העזרו ברקורסיה.



- נשים לב ששורש העץ מופיע בתחילת pre-order.
- ע"פ סריקת in-order כל האיברים שמופיעים לשורש חייבים להיות גם לימינו בעץ.
- כתע נוכל לחזור על הפעולה על תתי העץ עד שנשחזר את כל העץ.



PreOrder(x):

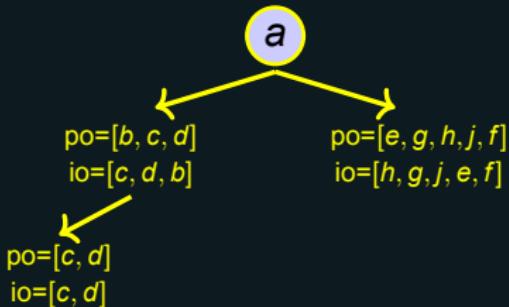
```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

.*a, b, c, d, e, g, h, j, f* :pre-order
.*c, d, b, a, h, g, j, e, f* :in-order
רמז: העזרו ברקורסיה.



- נשים לב ששורש העץ מופיע בתחילת סריקת pre-order.
- ע"פ סריקת in-order כל האיברים שמופיעים לשורש חייבים להיות גם לימינו בעץ.
- כעת נוכל לחזור על הפעולה על תתי העץ עד שנחזור את כל העץ.



PreOrder(x):

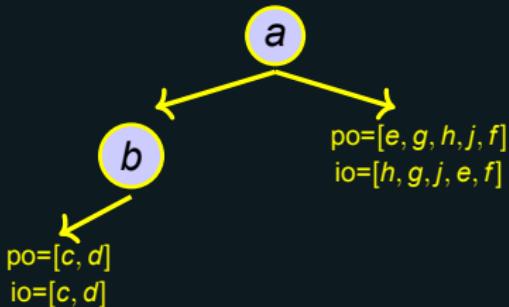
```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

.*a, b, c, d, e, g, h, j, f* :pre-order
.*c, d, b, a, h, g, j, e, f* :in-order
רמז: העזרו ברקורסיה.



- נשים לב ששורש העץ מופיע בתחילת pre-order.
- ע"פ סריקת in-order כל האיברים שמופיעים לשורש חייבים להיות גם לימינו בעץ.
- כעת נוכל לחזור על הפעולה על תתי העץ עד שנחזור את כל העץ.



PreOrder(x):

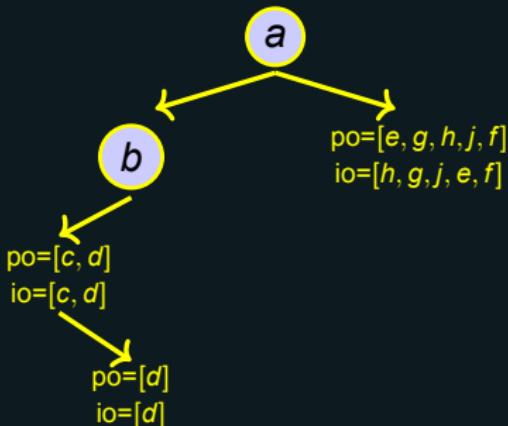
```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

.*a, b, c, d, e, g, h, j, f* :pre-order
.c, d, b, a, h, g, j, e, f :in-order
רמז: העזרו ברקורסיה.





PreOrder(x):

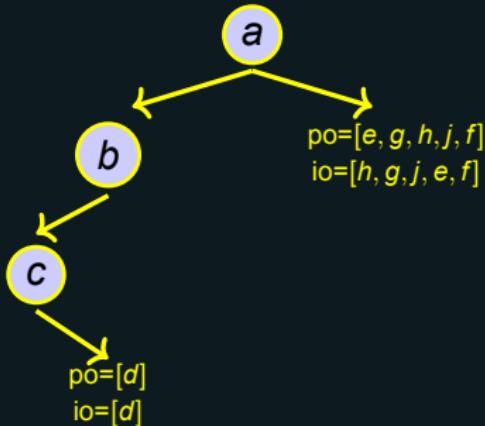
```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

.*a, b, c, d, e, g, h, j, f* :pre-order
.*c, d, b, a, h, g, j, e, f* :in-order
רמז: העזרו ברקורסיה.



- נשים לב ששורש העץ מופיע בתחילת pre-order.
- ע"פ סריקת in-order כל האיברים שמופיעים לשורש חייבים להיות גם לימינו בעץ.
- כעת נוכל לחזור על הפעולה על תת-העץ עד שנשחזר את כל העץ.



PreOrder(x):

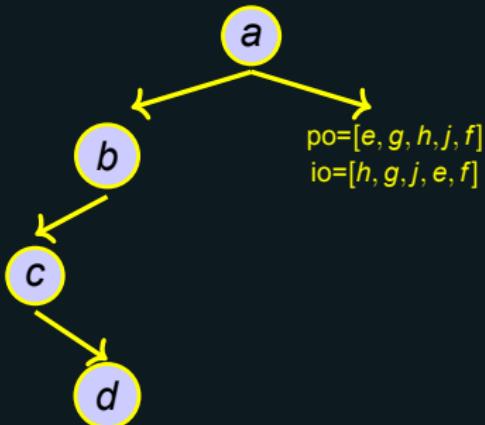
```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

.*a, b, c, d, e, g, h, j, f* :pre-order
.*c, d, b, a, h, g, j, e, f* :in-order
רמז: העזרו ברקורסיה.



- נשים לב ששורש העץ מופיע בתחילת pre-order.
- "עפ" סריקת in-order כל האיברים שמימיין לשורש חייבים להיות גם לימיין בעץ.
- כעת נוכל לחזור על הפעולה על תתי העץ עד שנשחזר את כל העץ.



PreOrder(x):

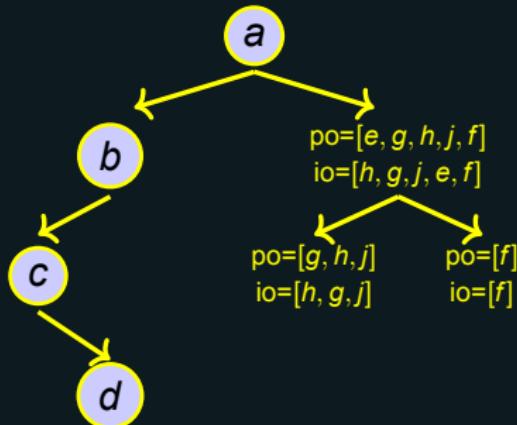
```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

.*a, b, c, d, e, g, h, j, f* :pre-order
.*c, d, b, a, h, g, j, e, f* :in-order
רמז: העזרו ברקורסיה.



- נשים לב ששורש העץ מופיע בתחילת pre-order סריקת order-in כל האיברים שמיינן.
- ע"פ סריקת order-in לשורש חייבים להיות גם לימינו בעץ.
- כתע נוכל לחזור על הפעולה על תתי העץ עד שנשחזר את כל העץ.

PreOrder(x):

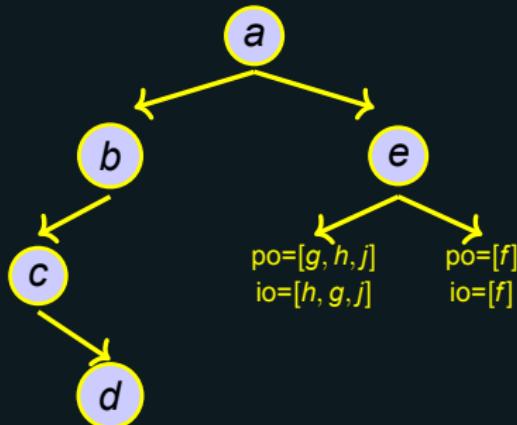
```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

.*a, b, c, d, e, g, h, j, f* :pre-order
.*c, d, b, a, h, g, j, e, f* :in-order
רמז: העזרו ברקורסיה.



- נשים לב ששורש העץ מופיע בתחילת pre-order סריקת.in-order
- ע"פ סריקת.in-order כל האיברים שמיינן לשורש חייבים להיות גם לימינו בעץ.
- כעת נוכל לחזור על הפעולה על תתי העץ עד שנשחזר את כל העץ.

PreOrder(x):

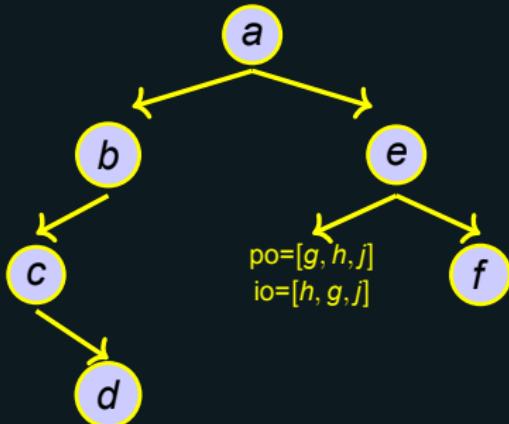
```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

.*a, b, c, d, e, g, h, j, f* :pre-order
.*c, d, b, a, h, g, j, e, f* :in-order
רמז: העזרו ברקורסיה.



- נשים לב ששורש העץ מופיע בתחילת pre-order סריקת order-in כל האיברים ממשמעו לעפ' סריקת order-in לשורש חייבים להיות גם לימים בעץ.
- cut נוכל לחזור על הפעולה על תתי העץ עד שנשחזר את כל העץ.

PreOrder(x):

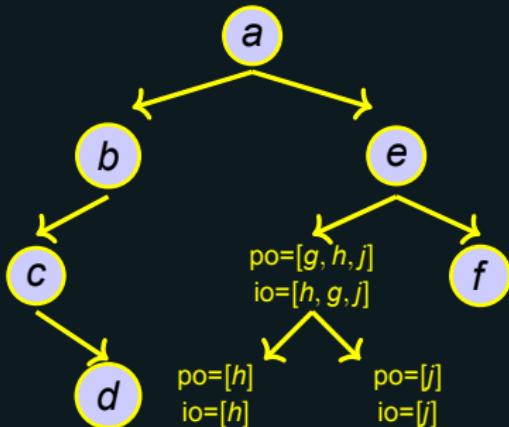
```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

.*a, b, c, d, e, g, h, j, f* :pre-order
.*c, d, b, a, h, g, j, e, f* :in-order
רמז: העזרו ברקורסיה.



- נשים לב ששורש העץ מופיע בתחילת pre-order סריקת order-in כל האיברים ממשמעו לשורש חייבים להיות גם לימינו בעץ.
- ע"פ סריקת order-in כל האיברים ממשמעו לשורש חייבים להיות גם לימינו בעץ.
- כתעת נוכל לחזור על הפעולה על תתי העץ עד שנחזור את כל העץ.



PreOrder(x):

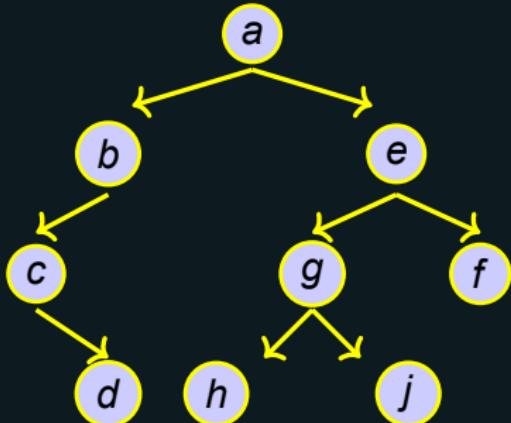
```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

InOrder(x):

```
if x.left ≠ Ø then InOrder(x.left)
Print x.key
if x.right ≠ Ø then InOrder(x.right)
```

להלן שתי סריקות של עץ חיפוש בינארי מסוים. שחזרו את העץ המקורי.

.*a, b, c, d, e, g, h, j, f* :pre-order
.*c, d, b, a, h, g, j, e, f* :in-order
רמז: העזרו ברקורסיה.



- נשים לב ששורש העץ מופיע בתחילת pre-order סריקת.
- ע"פ סריקת in-order כל האיברים שמופיעים לשורש חייבים להיות גם לימינו בעץ.
- icut נוכל לחזור על הפעולה על תתי העץ עד שנשחזר את כל העץ.



PreOrder(x):

```
Print x.key  
if x.left ≠ Ø then PreOrder(x.left)  
if x.right ≠ Ø then PreOrder(x.right)
```

הוכיחו / הפריכו את הטענה כי ניתן לשחזר עץ בהינתן סדריקות post-order ו pre-order

PostOrder(x):

```
if x.left ≠ Ø then PostOrder(x.left)  
if x.right ≠ Ø then PostOrder(x.right)  
Print x.key
```



PreOrder(x):

```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

הוכיחו / הפריכו את הטענה כי ניתן לשחזר עץ בהינתן סדריקות post-order ו pre-order.

PostOrder(x):

```
if x.left ≠ Ø then PostOrder(x.left)
if x.right ≠ Ø then PostOrder(x.right)
Print x.key
```

הטענה לא נכונה, נפריך באמצעות דוגמה.

PreOrder(x):

```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

הוכיחו / הפריכו את הטענה כי ניתן לשחזר עץ בהינתן סדריקות post-order ו pre-order.

PostOrder(x):

```
if x.left ≠ Ø then PostOrder(x.left)
if x.right ≠ Ø then PostOrder(x.right)
Print x.key
```

הטענה לא נכונה, נפריך באמצעות דוגמה.

בהינתן סדריקות:

a,b = pre-order

b,a = post-order

PreOrder(x):

```
Print x.key
if x.left ≠ Ø then PreOrder(x.left)
if x.right ≠ Ø then PreOrder(x.right)
```

הוכיחו / הפריכו את הטענה כי ניתן לשחזר עץ בהינתן סדריקות post-order ו pre-order.

PostOrder(x):

```
if x.left ≠ Ø then PostOrder(x.left)
if x.right ≠ Ø then PostOrder(x.right)
Print x.key
```

הטענה לא נכונה, נפריך באמצעות דוגמה.

בהינתן סדריקות:

a,b = pre-order

b,a = post-order

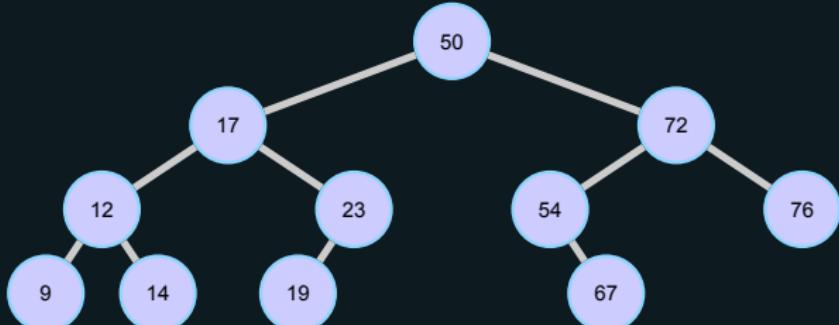
לא ניתן לדעת אם b הוא בן ימני או בן שמالي של a.

לפיכך קיימים שני עצים שונים שיתאימו לאותן סדריקות ולא ניתן להבחין ביניהם בעזרת.

תרגיל 4



נתון עץ חיפוש בינארי אשר בכל צומת שלו מתווד גם מצביע לאב (ק). כתבו תוכנית המקבלת עץ צומת w הממצירה את הצומת העוקב (הצומת בעל המפתח הקטן ביותר בגודל מהמפתח של w). מה הסיבוכיות של התוכנית?

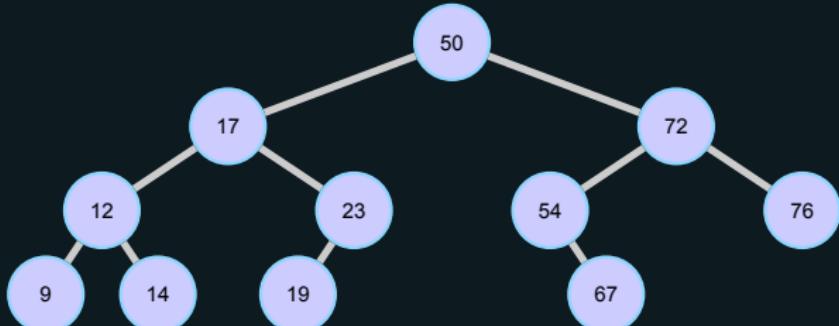


תרגיל 4



נתן עץ חיפוש בינארי אשר בכל צומת שלו מתווד גם מצביע לאב (ק). כתבו תוכנית המקבלת עץ צומת w המחזירה את הצומת העוקב (הצומת בעל המפתח הקטן ביותר הגובל מהמפתח של w). מה הסיבוכיות של התוכנית?

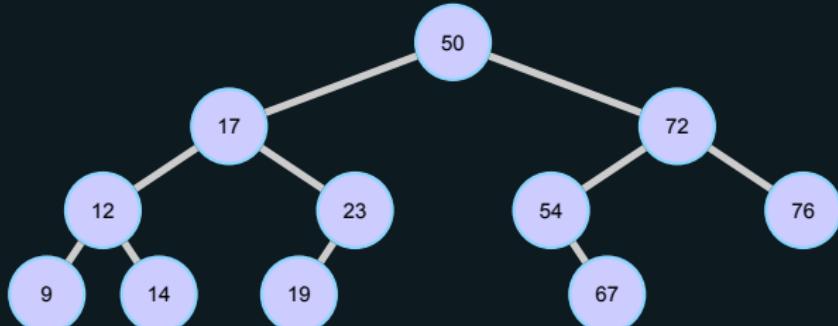
1. אם יש לצומת בן ימני, הצומת העוקב יהיה הקטן ביותר מתחת הימני. השתמש בתרגיל 1.





נתון עץ חיפוש בינארי אשר בכל צומת שלו מתווד גם מצביע לאב (ק). כתבו תוכנית המקבלת עץ צומת w המחזירה את הצומת העוקב (הצומת בעל המפתח הקטן ביותרგור בגודל מהמפתח של w). מה הסיבוכיות של התוכנית?

1. אם יש לצומת בן ימני, הצומת העוקב יהיה הקטן ביותר מתחת לעץ הימני. נשתמש בתרגיל 1.
2. אם אין לצומת בן ימני אז הצומת העוקב יהיה האב קדמון הצער ביותר של w שהבן השמאלי שלו הוא גם אב קדמון של w . נעלם אם כן במעלה העץ עד שהתנאי יתקיים.





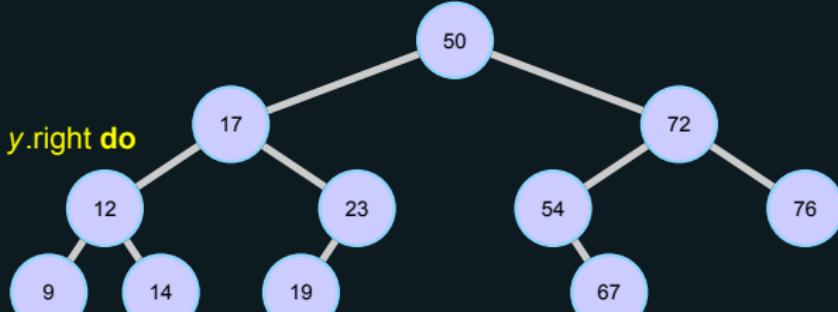
נתון עץ חיפוש בינארי אשר בכל צומת שלו מתווד גם מצביע לאב (ק). כתבו תוכנית המקבלת עץ צומת m המחזירה את הצומת העוקב (הצומת בעל המפתח הקטן ביותר הגדול מהמפתח של m). מה הסיבוכיות של התוכנית?

1. אם יש לצומת בן ימני, הצומת העוקב יהיה הקטן ביותר בתת העץ הימני. נשתמש בתרגיל 1.
2. אם אין לצומת בן ימני אז הצומת העוקב יהיה האב קדמון הצער ביותר של m שהבן השמאלי שלו הוא גם אב קדמון של m . נעלם אם כן במעלה העץ עד שהתנאי יתקיים.
3. אם התנאי לא התקיים הגיענו מימין ולכן התחלנו באיבר המקסימלי, כלומר אין צומת עוקב

`Successor(m):`

```

if x.right ≠ ∅ then
    return Min(x.right)
else
    y = m.p
    while y ≠ ∅ and m = y.right do
        m = y
        y = m.p
    if m = y.left then
        return y.key
    else
        return ∅
    
```





נתון עץ חיפוש בינארי אשר בכל צומת שלו מתווד גם מצביע לאב (ק). כתבו תוכנית המקבלת עץ צומת m המחזירה את הצומת העוקב (הצומת בעל המפתח הקטן ביותר הגדול מהמפתח של m). מה הסיבוכיות של התוכנית?

1. אם יש לצומת בן ימין, הצומת העוקב יהיה הקטן ביותר בתת העץ הימני. נשתמש בתרגיל 1.
2. אם אין לצומת בן ימין אז הצומת העוקב יהיה האב קדמון הצער ביותר של m שהבן השמאלי שלו הוא גם אב קדמון של m . נעלם אם כן במעלה העץ עד שהתנאי יתקיים.
3. אם התנאי לא התקיים הגיענו מימין ולכן התחלנו באיבר המקסימלי, כלומר אין צומת עוקב

`Successor(m):`

```

if x.right ≠ ∅ then
    return Min(x.right)
else
    y = m.p
    while y ≠ ∅ and m = y.right do
        m = y
        y = m.p
    if m = y.left then
        return y.key
    else
        return ∅
    
```

