

עצי חיפוש בינאריים כולל עצי AVL – תרגילים

שאלה 1

תארו אלגוריתם המקבל m עצי חיפוש בינאריים שבכל אחד מהם n איברים, ומאחד אותם לעץ חיפוש בינארי אחד.
סיבוכיות זמן הריצה הנדרשת היא $O(n \lg m)$.

שאלה 2

עליכם לתכנן מבנה נתונים לאחסון ותחזוקת קבוצת רשומות בעלות שני מפתחות. לכל רשומה R נציין ב- $key1[R]$ את המפתח הראשון וב- $key2[R]$ את המפתח השני. ידוע שהמספר הכולל של רשומות R בקבוצה הינו N ; מספר הערכים השונים של המפתח הראשון $key1[R]$ הוא n_1 ומספר הערכים השונים של המפתח השני $key2[R]$ הוא n_2 .

הציעו מבנה נתונים S לאחסון כל רשומות הקבוצה, שבאמצעותו ניתן לממש כל אחת מהפעולות הבאות בזמנים הנדרשים:

1. $SEARCH(k_1, k_2, S)$: חיפוש ב- S אחר רשומה כלשהי R בעלת זוג המפתחות (k_1, k_2) ; זמן הריצה $O(\lg n_1 + \lg n_2)$.
2. $INSERT(k_1, k_2, S)$: הכנסה ל- S של רשומה כלשהי R בעלת זוג המפתחות (k_1, k_2) ; זמן הריצה $O(\lg n_1 + \lg n_2)$.
3. $DELETE(p, S)$: מחיקה מ- S של הרשומה R , שאליה מצביע p ; זמן הריצה $O(\lg n_1 + \lg n_2)$.
4. $MAX2(S)$: מציאה, בין כל הרשומות R של S , של הערך המכסימלי של המפתח השני $key2[R]$; זמן הריצה $O(1)$.

שאלה 3

תארו אלגוריתם המקבל מצביע לשני עצים בינאריים t_1 ו- t_2 , ומחזיר $true$ אם ורק אם ניתן להפוך את t_1 ל- t_2 בעזרת רוטציות בלבד.

זמן הריצה הנדרש של האלגוריתם $O(n)$, כאשר n הוא מספר הצמתים הכולל בשני העצים.

שאלה 4

הוכיחו או הפריכו:

כאשר מבטלים שני איברים בעץ חיפוש בינרי, סדר ביטול האיברים אינו משפיע על מבנה העץ המתקבל.

שאלה 5

נניח שבעץ חיפוש בינרי מאוכסנים מספרים בין 1 ל-1000, וברצוננו לחפש את המספר 363. אילו מבין הסדרות הבאות **אינן** יכולות להיות סדרות של צמתים שנבחנו במהלך החיפוש? (קראו משמאל לימין).

1. 2, 252, 401, 398, 330, 344, 397, 363

2. 924, 220, 911, 244, 898, 258, 362, 363

3. 925, 202, 911, 240, 912, 245, 363

4. 2, 399, 387, 219, 266, 382, 381, 278, 363

5. 935, 278, 347, 621, 299, 392, 358, 363

שאלה 6

כתבו אלגוריתם המקבל עץ חיפוש בינרי וערך x , ומחזיר מצביע לצמת שערכו קרוב ביותר לערך x .

שאלה 7

כתבו אלגוריתם המקבל עץ חיפוש בינרי וערך x , ומחזיר מצביע למופע הראשון של x בעץ (המופע הראשון הוא המופע הראשון שייסרק בסדר תוכי). אם x לא בעץ יוחזר `nil`.

שאלה 8

תארו אלגוריתם המקבל עץ חיפוש בינרי וערך x , ומחזיר את המופע של x הקרוב ביותר אל השורש.
אם x לא בעץ, יוחזר nil .

שאלה 9

תארו אלגוריתם המקבל עץ חיפוש בינרי וערך x , ומחזיר את המופע של x שייסרק ראשון בסריקה
בסדר תחילי. אם x לא בעץ, יוחזר nil .

שאלה 10

תארו אלגוריתם שמקבל עץ AVL וערך x , ומוחק מהעץ את כל המופעים של x . זמן הריצה של
האלגוריתם $O(\lg n)$.

שאלה 11

נתאר אלגוריתם חלופי עבור מחיקת צומת z מעץ חיפוש בינרי.
במקרה השלישי, כאשר לצומת שני בנים, מאתרים את העוקב שלו y , ואז מחליפים בין $left[y]$
לבין $left[z]$; עכשיו אפשר להסיר את z כמו במקרה השני.
א. תארו יתרון אחד וחסרון אחד לפחות של אלגוריתם זה יחסית לאלגוריתם המחיקה המקורי
(המתואר בספר).
ב. האם ניתן להשתמש באלגוריתם החדש למחיקת צומת מעץ AVL?

שאלה 12

הציעו מבנה נתונים המאפשר לבצע את הפעולות הבאות בזמנים הנדרשים:
 $INSERT(S, z)$: הכנסת איבר בעל המפתח z למבנה S ; זמן: $O(\lg n)$
 $DELETE(S, p)$: מחיקת האיבר שאליו מצביע p מהמבנה S ; זמן: $O(\lg n)$
 $SUCCESSOR(S, z, k)$: מציאת העוקב ה- k של האיבר z במבנה S ; זמן: $O(\lg n)$.
הסבר: העוקב ה- k של z הוא העוקב של העוקב (k פעמים) של z .

שאלה 13

- א. נתון מערך ממורכז A בגודל n .
כתבו אלגוריתם רקורסיבי בשם BUILD-TREE, אשר יבנה מאיברי המערך עץ חיפוש בינרי מאוזן.
זמן הריצה של האלגוריתם צריך להיות $O(n)$.
ב. נתונים שני עצי חיפוש בינריים. כל אחד מהעצים מכיל n איברים.
ברצוננו להעביר את כל $2n$ האיברים לעץ חיפוש בינרי מאוזן (עץ החיפוש המאוזן ההתחלתי הוא ריק).
הסבירו איך ניתן לבצע את המשימה בזמן $O(n)$.

שאלה 14

A "super AVL tree" is an AVL tree with a pointer to the maximum key and a pointer to the minimum key in the tree.

- Explain how these pointers can be updated after insertion and deletion of nodes. The running times of these operations should remain $O(\lg n)$.
- It seems that a "super AVL tree" can do much more than a heap. Give two operations (that were discussed in the MOOC) that can be implemented on a "super AVL tree" more efficiently than on a heap.
- Give one operation (that was discussed in the MOOC) that can be implemented on a heap more efficiently than on a "super AVL tree".
-

שאלה 15

Write a function that gets an array of integers a of size n , which represents a complete binary tree and prints the pre-order-traversal of the tree.

The header of the function is:

void preorder (int a[], int n)

The running time is $O(n)$.

שאלה 16

Write a C function that receives an array of integers a of size n , which represents a complete binary tree and prints the post-order-traversal of the tree.

The header of the function is:

void postorder (int a[], int n)

The running time is $O(n)$.

שאלה 17

Let p be a node in a balanced binary search tree. The key of p is k . Write, in pseudocode, an algorithm that returns true if and only if p is the only node with key k . The running time of the algorithm is $O(\lg n)$, where n is the number of nodes in the tree.

שאלה 18

- Draw the highest balanced binary tree with 12 nodes.
- Draw the balanced tree of height 5 (the root's level is 0) with minimal number of nodes.

שאלה 19

t is the root of a binary tree. Write an algorithm (in pseudocode) that returns the height of the tree if the tree is balanced and -1 if it is not balanced. The running time of the algorithm is $O(n)$, where n is the number of nodes in the tree.

שאלה 20

Let p be a node in a binary tree. Write, in pseudocode, an algorithm that finds the successor of p in pre-order-traversal.

שאלה 21

Is the function *remove from a binary search tree* a commutative function?

In other words: does removing node x and then removing node y from a binary search tree always gives the same tree we would get from removing y first and then removing x ?

Prove your answer.