Exam 16 Dec 2021

Due No due date **Points** 60 **Questions** 4
Available until Dec 16, 2021 at 11:55am **Time Limit** 100 Minutes

Instructions

Attention! If an answer consists of a sequence of items, the items must be separated with commas, with no blank. No sign is to be put at the ends of the sequence.
For example: 1,3,4,6
Another example: APPLE,ORANGE,BANANA

This quiz was locked Dec 16, 2021 at 11:55am.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	99 minutes	20.54 out of 60

Correct answers are no longer available.

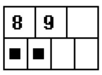
Score for this quiz: **20.54** out of 60
Submitted Dec 16, 2021 at 11:49am
This attempt took 99 minutes.

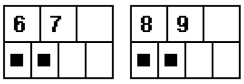
Partial

Question 1

7.5 / 15 pts

Given B+ tree **{[(2,4) 8 (9,10,12)] 14 [(14,15) 25 (25,27)]}** of degree 4.
Answer the questions. Use the following notations.

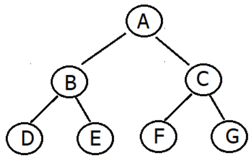
1. When you present a single node, give the keys in order separated by commas, without brackets. Do not use blanks!
   (Example: 

   this node is given like: 8,9)
2. When you present more nodes, put the nodes into brackets, the keys in order separated by commas. Do not use blanks!
   (Example: 

   these nodes are given like: (6,7)(8,9))

A) Draw the B+ tree above. The shape of that tree is given below where the letters identify the nodes. Present the nodes based on your drawing.

B: 8



F: 14,15

B) Illustrate insertion of 8. The following questions refer to the B+ tree which is the result of this insertion. Answer the questions.

- Present to root node. (See the 1st example.) 14
- Present the nodes at level one. (See the 2nd example.) 8,25
- Present the third leaf of the B+ tree. (See the 1st example. The nodes are indexed from 1, in left-to-right direction.) 14,15

C) Illustrate deletion of 25 from the original tree, which is given in textual form, in the first line of this whole exercise! Answer the following questions. The following questions refer to the B+ tree which is the result of this deletion.

1. The height of the B+ tree = 2
2. Present the root node (See the 1st example.) 8,14
3. Present the third leaf of the B+ tree. (See the 1st example. The nodes are indexed from 1, in left-to-right direction.) 12,14

Answer 1:

8

Answer 2:

14,15

Answer 3:

14

Answer 4:

8,25

Answer 5:

14,15

Answer 6:

2

Answer 7:

8,14

Answer 8:

12,14

Partial

Question 2

5 / 15 pts

Build an AVL tree with the following keys, i.e. insert them in order into an initially empty AVL tree.

18,4,15,2,20,1,12,3,9

1. How many times we apply balancing rotations during this process?
   3
2. Consider the resulting AVL tree. What is the balance of its root (+,-,=) ?
   +
3. Present the second level of the tree, each key followed by the balance of its node. (Remember that the root of a tree is at level zero. Example: 3=,5-,8+ (Do not use blanks.)) 2-,18=
4. Provided that we delete the root of this tree, Which key goes into the root of the new tree? 18
5. What is the balance of its root (+,-,=)? +
6. Delete the root of the new tree. Present the *level order* traversal of the resulting AVL tree. (Present only the appropriate sequence of the keys, without the balances. Do not use blanks. Example: 3,2,7)
   4,2,12,1,3,9,20

Answer 1:

3

Answer 2:

+

Answer 3:

2-,18=

Answer 4:

18

Answer 5:

+

Answer 6:

4,2,12,1,3,9,20

Partial

Question 3

3.75 / 15 pts

Consider the **Quick Search** algorithm and alphabet Σ= {*A, B, C, E*}.

We search for pattern *P* = ***BAB EBA***

in text *T* = ***EBABE BABEB ACBAB EBAEB***. Present the following values.

Neither T nor P contains spaces. The blanks only help reading! (Comparison means comparing the corresponding letters of T and P.)

- *shift(A)* = 1
- *shift(C)* = 7
- The size of the set of valid shifts, *S* = 4
- Present the smallest and the greatest valid shifts, in increasing order.
  S={ 123 , ..., 4 }
- During this searching for *P* in *T*, how many comparisons did the algorithm make *until* it has found the first valid shift? 6
- During this searching for *P* in *T*, how many comparisons did the algorithm make *after* it has found the first valid shift? 6
- During this searching for *P* in *T*, how many comparisons did the algorithm make altogether? 6

Answer 1:

1

Answer 2:

7

Answer 3:

4

Answer 4:

123

Answer 5:

4

Answer 6:

6

Answer 7:

6

Answer 8:

6

Partial

Question 4

4.29 / 15 pts

Select the true statements.

- [x] Consider a B+ tree with degree 6. In a leaf of it there may be maximum 6 pointers referring to data records of a direct file.

- [ ] Given a DAG of *n* vertices and *2n* non-negative edges where *n* is a large number. We have at hand optimized implementations of the *DAG shortest paths* and *Floyd-Warshall* algorithms. We want to find the shortest paths between all pairs of vertices of that graph. Running *n* times algorithm *DAG shortest paths* is more efficient than running *Floyd-Warshall* only once. (We can suppose that the central memory of the computer is large enough.)

- [ ] A (*p) node of a binary tree is balanced, iff for its balance: $p \rightarrow b < 2$.

- [x] In addition to initialization, the Floyd-Warshall algorithm (for a graph with no negative cycle) takes a total of n+n^2+n^3 iterations to determine the shortest paths between each pair of vertices.

- [ ] Only connected graphs have transitive closure.

- [x] Using Huffman coding, an encoded message can be easily split into smaller parts and decoded separately.

- [x] The inorder traversal of an AVL tree yields a strictly increasing sequence of keys.

- [ ] In an AVL tree the leaves are at the same depth in the tree.

- [ ] Given a B+ tree, and we insert some key. We can use only this key to expand the inner nodes of the tree.

- [x] Consider string matching and especially the KMP algorithm. One property of function *next* is the following one: *next(j+1) < next(j)+1 (j ∈ 1..m-1)*

- [x] Given a B+ tree. The keys in its leaves are strictly increasing in left-to-right order.

- [ ] Consider string matching and especially the Quick Search algorithm. Given a text of length 20 and a pattern of length 7. We will make at least 3 comparisons between the characters of the text and the characters of the pattern.

- [ ] $P[1..h+1] \sqsupset P[1..j+1] \Leftrightarrow P[1..h] \sqsupset P[1..j] \wedge P[h+1]=P[j+1]$

Given a graph of *n* vertices and *2n* non-negative edges where *n* is a large number. We have at hand optimized implementations of the *Dijkstra* and *Floyd-Warshal*l algorithms. We want to find the shortest paths between all pairs of vertices of that graph. Running *n* times algorithm *Dijkstra* is more efficient than running *Floyd-Warshall* only once. (We can suppose that the central memory of the computer is large enough.)

Given an AVL tree of *n* nodes. Searching for a key needs *O(log n)* time.

Quiz Score: **20.54** out of 60