CSPB 3104 - Park - Algorithms

Started on	Sunday, 18 February 2024, 3:52 PM
State	Finished
Completed on	Sunday, 18 February 2024, 3:57 PM
Time taken	5 mins 5 secs
Marks	29.00/29.00
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

When using the division method for hashing, our hash function has the form:

H(x) = x mod m

why do we normally choose a large prime number for m?

Select one:

a. Prime numbers are not a good choice for m.

b. They are guaranteed to maximize collisions

c. They are guaranteed to minimize collisions

d. They allow for many distinct slots with little chance of collisions

Your answer is correct.

The correct answer is: They allow for many distinct slots with little chance of collisions

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Assume that there are l slots in a hash table, where we are trying to load n elements. The load factor of the table is given by –

Select one:

a. n * l

b. l / n

c. l

d. n / l

Your answer is correct.

The correct answer is:
n / l

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 3.00 out of 3.00

Consider the hash function for hashing into a table with 100 "slots" labelled from 0 to 99:

$h_1(k) : k^2 \mod 100$

where k is the key. We will assume that the keys range from 0 to 999.

(A) Which of the following keys will have a hash collision with the key $k = 0$? If multiple answers seem correct, choose any one of them.

- (•) Any key with last digit 0 Correct
- ( ) 21
- ( ) 41
- ( ) 201
- ( ) 55

Mark 1.00 out of 1.00

The correct answer is: Any key with last digit 0

(B) Suppose numbers of the form ``x00" and ``x01" are quite common as keys. Which of the following problems will occur as a result? If multiple answers seem correct, choose any one of them.

- (•) All keys ending in 0 will collide on to the slot 0 Correct
- ( ) All keys ending in 01 will collide on to slot 1
- ( ) All such keys will collide onto the same slot
- ( ) No problems, the hash table will handle such keys just fine
- ( ) There is no slot in the table for such keys to be stored

Mark 1.00 out of 1.00

The correct answer is: All keys ending in 0 will collide on to the slot 0

(C) Which of the following slots will always go **unoccupied** in the hash table regardless of which key we insert? If multiple answers seem correct, choose any one of them.

- (•) 7 Correct
- ( ) 3
- ( ) 25
- ( ) 16
- ( ) 49

Mark 1.00 out of 1.00

The correct answer is: 7

Correct

Marks for this submission: 3.00/3.00.

Question **4**

Correct

Mark 3.00 out of 3.00

Now consider the hash function:

$h_2(k) : k \mod 101$

that hashes keys k from the range $0, \ldots, 999$ to slots in a table of size 101.

(A) How many keys collide with $k = 0$? (do not include 0 itself in your count). Enter the number in the space below.

9

(B) How many unused slots exist in the hash table? An unused slot is one where no key can hash into it. Enter the number in the space below.

0

(C) If all possible keys from 0 to 999 were inserted into the table. What is the maximum number of keys on any single slot in the table? Enter the number in the space below.

10

Correct

Marks for this submission: 3.00/3.00.

Question **5**

Correct

Mark 13.00 out of 13.00

Consider any given red-black tree T with n internal nodes. Let r denote the root of the tree and bh(r), denote the blackheight of the root. Answer TRUE/FALSE to each of the questions below.

[S0] The height of the tree is $O(\log_2(n))$.

TRUE

[S1] It is possible to construct red-black trees of height $n-1$ in general, for all values of n.

FALSE

[S2] Any path from the root to a leaf must have *at least bh(r) nodes* (not counting the root but counting a leaf).

TRUE

[S3] Any path from the root to a leaf must have *at most 2*bh(r) nodes*.

TRUE

[S4] Any path from root to leaf has **exactly bh(r) black nodes** (not counting the root, but counting the leaf).

TRUE

[S5] Any path from root to a leaf has **at least bh(r) red nodes**.

FALSE

[S6] It is possible for the tree to have **no red nodes**.

TRUE

[S7] It is possible for a red-black tree to have **no black nodes**.

FALSE

[S8] A red node can have a non-leaf left child and a leaf (nil) right child.

FALSE

[S9] Every path from the root to a leaf has the same length.

FALSE

[S10] If a new node is inserted using the algorithm described in CLRS, it is always inserted at a leaf as a red node.

TRUE

[S11] Inserting a new node as a red node in the tree, may cause an imbalance in the number of black nodes from root to the leaf.

FALSE

[S12] If a new node is inserted in the tree as a red node, $\Theta(\log_2(n))$ tree rotation operations can restore the red-black property.

TRUE

Correct

Marks for this submission: 13.00/13.00.

Question **6**

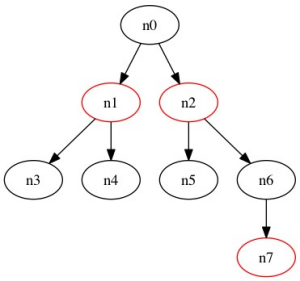
Correct

Mark 4.00 out of 4.00

This problem involves the consequences of the four basic rules of a red-black tree, recalled below:

1. Every node must be red or black.
2. The root must be black and the leaves must be labelled nil, and colored black
3. Both the children of a red node are black.
4. For any node n, all paths from n to its descendent leaves must encounter the same number of black nodes.

Consider tree T1 shown below



The leaves "nil" are not shown in the diagram.

(A) The tree above is a valid red-black tree satisfying the properties listed above? Write TRUE/FALSE in the space below.

TRUE

(B) What is the height of the tree? Enter the number in the space below.

4

(C) What is the black height of the root **n0** of the tree? Enter the number in the space below.

2

(D) Which of the following changes to the tree will **NOT** invalidate the red-black properties?

- (•) Changing nodes n1 and n2 to both be black Correct
- ( ) Changing the color of node n7 to black
- ( ) Changing the color of n5 to red
- ( ) Making all nodes in the tree black
- ( ) None of the changes can be made in the list can be made.

Mark 1.00 out of 1.00

The correct answer is: Changing nodes n1 and n2 to both be black

Correct

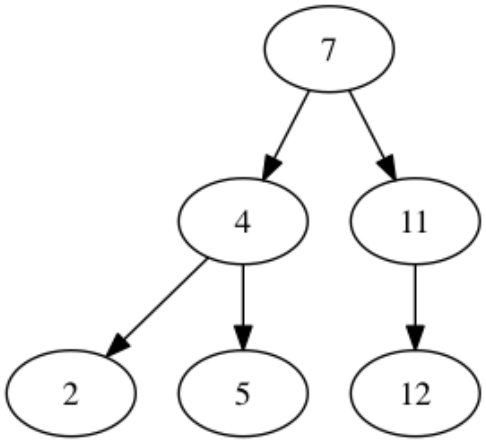
Marks for this submission: 4.00/4.00.

Question **7**

Correct

Mark 3.00 out of 3.00

Consider the tree below:



Suppose we wish to right rotate the tree with $x = 4$ and $y = 7$, so that the new root is 4.

Write down the numbers corresponding to the nodes below in the tree after rotation:

The right child of the new root (4) will be:

7

The left child of 7 in the new tree will be:

5

Which of the following remain the same during the right rotation?

- (●) Left subtree of 4 and right subtree of 7 Correct
- ( ) Right subtree of 4 and right subtree of 7
- ( ) Left subtree of 4 and left subtree of 7
- ( ) Right subtree of 4 and left subtree of 7

Mark 1.00 out of 1.00

The correct answer is: Left subtree of 4 and right subtree of 7

Correct

Marks for this submission: 3.00/3.00.

Question **8**

Correct

Mark 1.00 out of 1.00

In a hashing system that uses chaining to resolve conflicts, what is the worst-case complexity of a lookup ? (assume the elements are chained into an unordered list)

Select one:

a. O(log n)

b. O(1)

c. O(n^2)

d. O(n)

Your answer is correct.

The correct answer is: O(n)

Correct

Marks for this submission: 1.00/1.00.