Module 3 Graded Assignment and Quiz

Due Sep 24, 2021 at 11:59pm **Points** 6 **Questions** 6
Available Sep 11, 2021 at 12am - Feb 11 at 11:59pm 5 months
Time Limit None

Instructions

Homework Prompt

[Q1]

A sequence of n operations is performed on a data structure. The ith operation costs i if i is an exact power of 2, and 1 otherwise. That is, operation i costs f(i), where f(i) = i, if $i = 2^k$ for some $k \geq 0$, and f(i) = 1 otherwise. Determine the amortized cost per operation using the following methods of analysis:

(a) Aggregate method

(b) Accounting method

(c) Potential method

[Q2]

Suppose that we have a stack S that supports the following operations: Push (S,x), which pushes an element x on top of the stack S, Pop (S), which removes the element at the top of stack S, and MultiPop (S,k), which removes the top k elements from the stack S (if we try to remove more elements than the current number of elements on the stack in a Pop or MultiPop operation, only the elements currently on the stack will be returned). Assume that the stack is initially empty. Give a potential function argument showing that the amortized running time of each of the three stack operations is Θ(1).

[Q3]

Consider an ordinary binary heap data structure with n elements that supports the instructions Insert and Extract-Min in O(log n) worst- case time. Give a potential function Φ such that the amortized cost of Insert is

O(log n) and the amortized cost of Extract-Min is O(1), and show that it works.

[Q4]

The following binary search tree (BST) operations can be implemented efficiently on splay trees. For simplicity, assume that all elements are distinct.

Join (T1, T2): Assume every element in BST T1 is less than or equal to every element in BST T2, and returns the BST formed by combining the trees T1 and T2.

Split (T,a): Split BST T, containing element a, into two BSTs: T1, containing all elements in T with key less or equal to a; and T2, containing all elements in T with key greater than a.

Show how to correctly implement those operations on splay trees so that the amortized running time of your operations is O(log n). Justify your procedures and their corresponding running times.

Would you be able to also implement the Join and Split operations on a Red-Black tree in O(log n) worst-case running time? Briefly justify your answer.

[Q5]

We define an augmented binary search tree to be a binary search tree where each node v in the tree also stores the value size[v], which is equal to the number of nodes in the subtree rooted at v.

(a) Show that a rotation in an augmented binary tree can be performed in constant time.

(b) Describe an algorithm SplaySelect (k), $1 \leq k \leq n$, that selects the kth smallest item in an augmented splay tree with n nodes in O(log n) amortized time. Justify your algorithm and its running time.

[Q6]

Show the result of inserting the values 2, 1, 4, 5, 9, 3, 6, 7 into an empty splay tree. Show the tree at the end of each insertion. Show each rotation.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	1,278 minutes	6 out of 6

Score for this quiz: **6** out of 6
Submitted Sep 23, 2021 at 12:01pm
This attempt took 1,278 minutes.

Question 1

1 / 1 pts

Regarding homework question 1, give the potential function necessary to determine the amortized time complexity of each of the *n* operations using the potential method. Let *i* be the ith iteration and assume the potential function is equal to zero in its initial state.

- $\phi(i) = 2i$

Correct!

- $\phi(i) = 0$ if $i = 0$, and $\phi(i) = 2i - 2^P$ where 2^P is the next largest power of 2 strictly greater than i otherwise
- $\phi(i) = 2^i$
- $\phi(i) = 0$ if $i = 0$, and $\phi(i) = 2i - 2^P$ where 2^P is the next largest power of 2 greater than or equal to i otherwise

Question 2

1 / 1 pts

Regarding homework question 2, which of the following describes the amortized time of the ith operation on the stack data structure?

- The size of the stack at iteration i minus i plus the cost

The size of the stack after the ith iteration plus the operation cost

Correct!

The difference in the size of the stack between iterations i and $i-1$ plus operation cost

Two times the difference in the size of the stack between iterations i and $i-1$ plus operation cost

Question 3

1 / 1 pts

Regarding homework question 3, identify the potential function required for operations on a binary heap.

The branching factor of the binary heap

Correct!

The sum of node depths over the set of all nodes.

The height of the binary heap

The number of vertices in the binary heap

Question 4

1 / 1 pts

Regarding homework question 4, which of the following describes how a $Join(T_1, T_2)$ procedure would work in O(log(n)) time given the constraints on T_1, T_2?

Correct!

Let the maximum element of T_1 be denoted as v. Splay v and let the right child of v be the root node of T_2.

Let the maximum element of T_2 be denoted as v. Splay v and let the right child of v be the root node of T_1.

Let the minimum element of T_1 be denoted as v. Splay v and let the left child of v be the root node of T_2.

Question 5

1 / 1 pts

Regarding question 5, which of the following pseudocodes describes how $SPLAYSELECT(k)$ would work given key k and a function $size(v)$ which returns the number of nodes in the subtree rooted at node v?
Note: ----> denotes an indent.

SPLAYSELECT(k)

------->SPLAY(SELECT(k,root-node))

SELECT(k,node v)

------->If size(v.left-child) = k-1 then return v

------->If size(v.left-child)+1 >= k

---------->then return SELECT(k-size(v.left-child)-1,v.left-child)

------->Else return SELECT(k,v.left-child)

Correct!

SPLAYSELECT(k)

------->SPLAY(SELECT(k,root-node))

SELECT(k,node v)

------->If size(v.left-child) = k-1 then return v

------->If size(v.left-child)+1 < k

---------->then return SELECT(k-size(v.left-child)-1,v.right-child)

------->Else return SELECT(k,v.left-child)

SPLAYSELECT(k)

------->SPLAY(SELECT(k,root-node))

SELECT(k,node v)

------->If size(v.left-child) = k-1 then return v

------->If size(v.left-child)+1 > k

---------->then return SELECT(size(v.left-child)-1,v.right-child)

------->Else return SELECT(k,v.left-child)

SPLAYSELECT(k)

------->SPLAY(SELECT(k,root-node))

SELECT(k,node v)

------->If size(v.left-child) = k-1 then return v

------->If size(v.left-child)+1 > k then return SELECT(k,v.right-child)

------->Else return SELECT(k,v.left-child)

Question 6

1 / 1 pts

Show the result of inserting the values 2, 1, 4, 5, 9, 3, 6, 7 into an empty splay tree. Show the tree at the end of each insertion. Show each rotation.

What is the height of the resulting tree? Recall that tree height is the number of edges along the path spanning the root node to the farthest leaf node.

7

3

Correct!

4

5

Quiz Score: **6** out of 6