MTH 4320 Homework 5

Yaohui Wu

March 13, 2024

Problem 1

Solution. Let *sum* be an attribute of the stack and *sum* is 0 when we create the stack. If the stack is empty and we push an element then we add that number to *sum* to update the sum of the stack. Similarly, if the stack is not empty and we pop an element then we subtract it from *sum* to update the sum. If S_1 is empty and we push an element then *max* is that element which we also push it to S_2. Let S_1 be the stack storing all of the elements and let S_2 be a stack that stores the values of *max*. Every time we push an element we compare it with *max* and update *max* if the new element is greater then push the new element to both S_1 and S_2. Similarly, when we pop an element from S_1 we check if it is the top element in S_2 which is *max*. If it is *max* then remove the element from S_1 and S_2. The new *max* of S_1 is the new top element in S_2. Since all operations take $O(1)$ time to update *sum* and *max*, hence the time complexity of the operations *sum* and *max* are $O(1)$. ■

Problem 2

Solution. We can implement the disjoint-set data structure using trees. We implement Create(x) by creating a new tree with a single vertex x and this takes $O(1)$ time. The root of the tree has the name of the set and number x. The leaves are all of the elements in the same set. We implement Find(x) by iterating over all vertices of all the trees so this takes $O(n)$ time. We implement Union(Y, Z) by attaching the root of Z to Y and update its name of the set to Y so the time complexity is $O(1)$. ■

Problem 3

Solution. The implementation of the data structure is

- Init(p): Create a linked list with p keys as roads and the value of each node is a linked list with the values as bridges. The time complexity is $O(p)$.
- Build(r, h): Insert a new element h to the linked list r.The time complexity is $O(1)$.
- Road(r): Traverse the linked list r to print every element h. The time complexity is $O(t)$.

- Trip(h): Traverse all linked lists for roads and print if there is an element greater than h in the linked list of bridges. The time complexity is $O(p)$.

■

Problem 4

Solution. Let G be a graph where the vertices represent the entries of the Sudoku grid. There is an edge between two vertices if they are in the same row or the same column. In addition, there is an edge between two vertices in the same subgrid. The graph coloring problem is to color the vertices with nine different colors s.t. no adjacent vertices have the same color. ■