Fundamentals of Data Structures & Algorithms

Online Open-Book Exam

Wednesday 2nd December 2020, 11 AM – 1.30 PM

Attempt ALL questions

Question 1

Consider the set of initially unrelated elements 0, 1, 2, 3, 4, 5, 6, 7.

Draw the forest of trees and the corresponding array that results from each of the following sequence of operations using the *quick find* algorithm.

union(0, 2), *union*(3, 4), *union*(6, 5), *union*(6, 7), *union*(4, 7).

What is the expected running time of the union operation in this case?

Question 2

Outline, using diagrams, the *Stack* (implemented both using a resizing array of initial size 2 and linked structure) that results from the following sequence of operations:

push(A), push(B), push(C), pop(), push(D), push(E), pop(), push(E), push(G)

Question 3

List (seven) common order of growth classification in order of increasing run-time for large *N* and give an example of a problem that corresponds to each classification.

Question 4

Briefly outline the *Selection Sort* algorithm.

Use your algorithms to sort the following array, showing a trace of the algorithms workings at each stage.

[S, O, R, T, E, X, A, M, P, L, E]

Question 5

Briefly outline the *Merge Sort* algorithm.

Use your algorithm to sort the following array, showing a trace of the algorithms workings at each stage.

[17, 22, 65, 91, 37, 52, 40, 12, 41]

Question 6

What does it mean for a sorting algorithm to be *stable* and why might this matter?

Of the sorting algorithms you have studied, list two which are stable and two which are not stable and for each one briefly show why this is the case.

Question 7

For a *Binary Heap*, for each scenario below, show using suitable diagrams how the heap order is restored. If exchanges are required, how is the location of the key to exchange with determined.

i. a child's key becomes larger key than its parent's key.
ii. a parent's key becomes smaller than one (or both) of its children's.

Question 8

Heap Sort is an efficient in-place sorting algorithm. The basic outline for heap sort is:

- Create a max-heap with all *N* keys.
- Repeatedly remove the maximum key.

How is the max-heap constructed? Demonstrate the process for the initial unsorted array below:

[S, O, R, T, E, X, A, M, P, L, E]

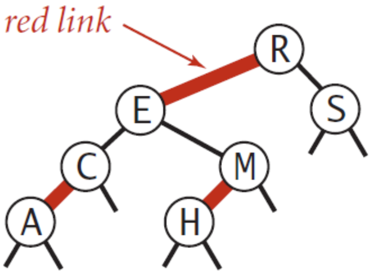
Question 9

Describe a *Binary Search Tree* (BST) and its properties with a suitable diagram.

Into your *Binary Search Tree*, *insert* two keys and *delete* the root key (in that order) demonstrating how the BST properties are maintained by each operation.

Question 10

Consider the following *Left-Leaning Red-Black Tree*. Add the key *Z*, then add the key *P*.



i. Draw the resulting left-leaning red-black tree.
ii. How many left rotations, right rotations, and colour flips are performed in total to insert the two keys?