Student Name: ______________________________ J#: ________________

CSC 228 Data Structures and Algorithms, Spring 2020
Instructor: Dr. Natarajan Meghanathan

Exam 3 (Take Home) Max. Points: 100 points

Due on: April 16th, by 11.59 PM in Canvas. **Late submissions will NOT be accepted.**

Submission Options (choose one of the three): You can either
(a) Print this exam, write the solutions in the space provided, scan and upload as a PDF file or
(b) Use the space provided to type the solutions, save the file to a word or PDF and upload or
(c) Write the solutions for some questions by hand and type the solutions for some other questions. In this case, you should scan the written text to a PDF file, merge it with the PDF file for the typed content and submit everything together as a single PDF file.

Clear label your answers for each of the sub parts (a) - (n), in the space provided.

Given an array of integers, do the following (SHOW ALL THE STEPS; just writing the final answer will get only ZERO).

(a - 11 pts) Construct a max heap of the array. Show the initial essentially complete binary tree and the transformation of the binary tree to a max heap via the reheapify operations at the indices of the internal nodes (as shown in the slides).

(b - 11 pts) Sort the max heap version of the array obtained from (a) to obtain a sorted array of integers. Show the structural changes of the max heap in each iteration.

(c - 7 pts) Transform the max heap of (a) to a binary search tree.

(d - 7 pts) For the binary search tree obtained in (c), considering both its structure and data together, determine the average number of comparisons for a successful search and the average number of comparisons for an unsuccessful search.

(e - 7 pts) Use the sorted array of (b) to construct a binary search tree.

(f - 7 pts) For the binary search tree obtained in (e), considering both its structure and data together, determine the average number of comparisons for a successful search and the average number of comparisons for an unsuccessful search.

(g - 6 pts) Construct a hash table of the given array using a hash function H(K) = K mod 5.

(h - 6 pts) For the hash table of (g), determine the average number of comparisons for a successful search and the worst case number of comparisons for an unsuccessful search.

(i - 9 pts) Consider the elements of the array assigned to you are known only one at a time. Construct a sequence of priority queues (as max heaps) with the insertion (enqueue) of one element at a time, as shown in the slides.

(j - 7 pts) Transform the binary search tree of (c) to a min heap.
(k - 6 pts) Transform the max heap of (a) to a min heap via reheapify operations at the internal nodes.

(l - 3 pts) Are the min heaps of (j) and (k) the same or not? What can you say about the distribution of the data in the min heaps of (j) and (k)?

(m - 7 pts) Determine whether the binary search trees of (c) and (e) are height-balanced or not?

(n - 6 pts) Write the pre-order, in-order and post-order traversal of the vertices in the binary search tree of (c).

Student Name: ______________________________ J#: ________________

1	Allen, Kenneth	24	44	31	29	19	45	47	36	41	42	25	35
2	Allen, Laura	26	37	45	6	46	29	48	6	37	20	32	29
3	Bracamonte, Piero	26	13	46	47	10	41	18	27	16	10	24	38
4	Coffee, Jhakorri	12	47	44	7	33	43	30	25	46	6	39	38
5	Green, Brandon	41	5	27	19	4	37	30	46	8	26	10	16
6	Harper, Antonio	41	13	15	5	9	42	2	17	31	40	10	30
7	Irving, Morenike	20	47	23	1	31	45	38	37	8	2	47	44
8	Jackson, Jamira	39	10	49	46	12	38	11	2	50	41	39	34
9	Johnson, Guy	9	36	33	21	35	33	4	28	22	35	48	44
10	Lenard, Jorian	38	15	22	45	14	9	11	41	43	16	33	50
11	Mcmorris, Dwayne	36	5	12	47	13	19	37	44	19	47	21	34
12	Moqbel, Basel	34	10	26	37	41	36	6	10	41	18	39	39
13	Price, Vincent	40	28	1	38	33	20	2	22	31	50	5	13
14	Robinson, Godfrey	21	33	28	46	32	5	3	39	36	4	40	47
15	Scott, Marlisa	42	42	41	10	46	27	18	43	11	24	13	47
16	Sims, Kandice	1	36	29	38	29	43	19	48	17	42	47	28
17	Slaughter, Toy	24	28	38	29	10	34	37	28	35	19	19	45
18	Thompson, Da Johne'	14	17	16	45	9	42	47	34	33	45	49	42
19	Varnell, Kenneth	44	10	46	10	10	12	22	24	33	26	44	47
20	Walls, Justin	29	48	19	10	43	1	37	14	18	30	8	5
21	Williams, Javian	17	6	43	20	31	4	45	33	15	15	30	38
22	Williams, Nicholas	36	7	28	45	44	50	48	6	26	9	49	32
23	Williams, Opeoluwa	39	2	1	16	31	21	1	27	13	10	20	37
24	Wilson, Kyla	7	1	12	39	34	7	5	26	19	22	5	20
25		7	6	16	33	29	8	39	36	16	5	34	28
26		37	5	6	12	7	8	24	12	23	39	3	7
27		26	4	48	28	15	15	22	1	2	50	29	49
28		16	13	22	42	17	3	22	32	30	18	12	38
29		4	8	16	6	6	35	11	40	38	16	33	42
30		30	44	21	41	8	35	33	20	46	29	16	32
31		25	42	35	11	25	41	45	21	10	25	27	23
32		8	41	13	21	2	44	3	1	18	45	49	14
33		48	19	9	1	44	25	48	14	19	9	11	46
34		17	39	41	35	48	29	43	46	34	32	14	49
35		50	14	34	1	34	21	29	30	38	8	31	35
36		43	50	45	44	20	9	8	36	50	5	13	24
37		35	49	15	19	23	1	18	22	42	13	30	45
38		44	21	16	26	34	41	33	10	7	18	38	35
39		19	41	36	8	24	23	50	38	14	25	4	16
40		49	45	6	5	21	13	48	14	21	14	33	8
41		15	4	37	23	10	36	46	23	44	29	49	14
42		39	35	7	36	26	29	40	17	21	48	26	20
43		12	48	10	5	38	34	3	25	15	3	47	37
44		13	1	10	2	2	37	17	19	39	29	32	30
45		20	4	21	6	49	35	45	17	45	46	1	47
46		45	46	15	24	39	9	13	1	35	43	15	32

(a - 11 pts) Construct a max heap of the array. Show the initial essentially complete binary tree and the transformation of the binary tree to a max heap via the reheapify operations at the indices of the internal nodes (as shown in the slides).

(b - 11 pts) Sort the max heap version of the array obtained from (a) to obtain a sorted array of integers. Show the structural changes of the max heap in each iteration.

(c - 7 pts) Transform the max heap of (a) to a binary search tree.

(d - 7 pts) For the binary search tree obtained in (c), considering both its structure and data together, determine the average number of comparisons for a successful search and the average number of comparisons for an unsuccessful search.

(e - 7 pts) Use the sorted array of (b) to construct a binary search tree.

(f - 7 pts) For the binary search tree obtained in (e), considering both its structure and data together, determine the average number of comparisons for a successful search and the average number of comparisons for an unsuccessful search.

(g - 7 pts) Construct a hash table of the given array using a hash function H(K) = K mod 5.

(h - 7 pts) For the hash table of (g), determine the average number of comparisons for a successful search and the worst case number of comparisons for an unsuccessful search.

(i - 14 pts) Consider the elements of the array assigned to you are known only one at a time. Construct a sequence of priority queues (as max heaps) with the insertion (enqueue) of one element at a time, as shown in the slides.

Student Name: ______________________________ J#: ________________

(j - 7 pts) Transform the binary search tree of (c) to a min heap.

(k - 7 pts) Transform the max heap of (a) to a min heap via reheapify operations at the internal nodes.

Student Name: ______________________________ J#: ________________

(l - 4 pts) Are the min heaps of (j) and (k) the same or not? What can you say about the distribution of the data in the min heaps of (j) and (k)?

(m - 7 pts) Determine whether the binary search trees of (c) and (e) are height-balanced or not? Show all the work.

(n - 6 pts) Write the pre-order, in-order and post-order traversal of the vertices in the binary search tree of (c).