

**COMP 2230 – Data Structures and Algorithm Analysis**

Assignment #9: Priority Queues and Hashing

## Due Date: Section 01 Nov. 21st Section 02 Nov 22nd, 2024

**Chapter 21**

1. Draw the priority queue (based on min heap) that results from adding the following integers: 4, 5, 31, 187, 6, 312, 11, 112, 7

A drawing of a network

Description automatically generated

1. Starting with the previous priority queue draw the Priority Queue that results from performing a remove operation.

A drawing of a diagram

Description automatically generated

1. Repeat question 1 and 2, this time with a priority queue (based on max heap).

A drawing of a diagram

Description automatically generatedA diagram of numbers and circles

Description automatically generated

**Appendix I**

Problem 1:

Using a fixed size array of size 17 insert the following numbers into the table using number % table size as the hashing function. Use linear probing to resolve collisions.

10, 20, 30, 40, 50, 60, 70, 80 , 90

Problem 2:

Using a fixed size array of size 17 insert the following numbers into the table using number % table size as the hashing function. Use linear probing to resolve collisions

11, 23, 31, 43 , 53 , 61, 79, 89 , 97

Problem 3:

Is there a difference in the collision rates for problem 1 and problem 2 above?

Problem 4:

Using a fixed size array of size 16 insert the following numbers into the table using number % table size as the hashing function. Use linear probing to resolve collisions.

10, 20, 30, 40, 50, 60, 70, 80, 90

Problem 5:

Using a fixed size array of size 16 insert the following numbers into the table using number % table size as the hashing function. Use linear probing to resolve collisions.

11, 23, 31, 43, 53, 61, 79, 89 , 97

Problem 6:

Is there a difference in the collision rates for problem 4 and problem 5 above?

Problem 7:

Write a program to check the hash index for all numbers between 1 and 1000 using a hash table of 17 and a hash table of size 16, this program is to count the distribution of hash keys generated for each table size. i.e. how many numbers may to key 1, 2, 3 etc for table size 17 and for table size 16. **What do your results tell you?**