

1. Algorithms, Running Time, Trees, Recursion and Sorting (60pts):

a. Answer: See Exercise1.java

2. Graphs Data Structure and Depth First Search (30pts):

a. Answer: See Exercise2.java

3. Binary Search Trees (BST) (40pts)

a. Answer: See Exercise3.java

4. Hash tables and Hash functions (40pts):

a. A Hash table (also known as a hash map) is a data structure that implements an associative array abstract data type structure that maps keys to values.

b. A Hash function is used by the hash table creation algorithm to take an original input and follow a set of rules to return a summary of that input to be added as an index into an array of buckets or slots from which the original value can be found. For example, with a string – a hash function could return its length modulo X, where X is an integer.

c. Collisions occur when different inputs into the hash function produce the same index output and be hashed to the same slot. For example strings AAA and BBB, while different, after being passed to a hash function that returns their lengths would both produce an index of 3.

d. The difference between a Hash table and a Hash map:

i. HashMap is non synchronized. It is not-thread safe and can't be shared between many threads without proper synchronization code whereas Hash table is synchronized. It is thread-safe and can be shared with many threads.

ii. HashMap allows one null key and multiple null values whereas Hash table doesn't allow any null key or value.

iii. HashMap is generally preferred over Hash table if thread synchronization is not needed To successfully store and retrieve objects from a Hash table, the objects used as keys must implement the hash function and the equals method. Since null is not an object, it can't implement these methods.

iv. Sources:

<https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/util/Hashtable.html>,

<https://www.geeksforgeeks.org/differences-between-hashmap-and-hashtable-in-java/>

e. Desired times for search, insert, delete algorithms in a Hashtable are:

	Insert (add)	Delete	Search
Chaining	O(1)	Worst case: O(n) , If hash function splits uniformly, the Best Case is: O(1) , Average: O(1)	Worst case: O(n) , If hash function splits uniformly, the Best Case is: O(1) , Average: O(1)
Probing	Worst case: O(n) , If hash function splits uniformly, the Best Case is: O(1) , Average: O(1)	Worst case: O(n) , If hash function splits uniformly, the Best Case is: O(1) , Average: O(1)	Worst case: O(n) , If hash function splits uniformly, the Best Case is: O(1) , Average: O(1)

Were n = number of elements you want to add, m = number of buckets. For chaining $n > m$ is ok, for probing n must be $\leq m$.

f. Ways to deal with collision:

- i. Chaining is a way of dealing with collision by storing all elements that hash to the same slot in a linked list, and entering the pointer to the head of the said list in the corresponding hash

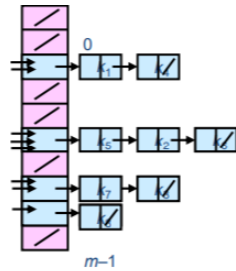
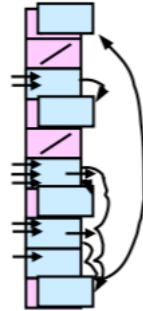


table slot:

- ii. Linear Probing (Open Addressing) is another way of dealing with collision, where all of the elements are stored in the hash table itself, but there is a systematic way (rule) of where to store an element if its intended place in the hash table is already taken by another element. One way, for example, is to always store the conflicting value in the immediate next free slot available.



g. Example of storing strings with hash function and hash table with linear probing:

Animals (Strings): Cat, Horse, Elephant, Wolf, Dog.

1. Cat -> Hash function (length) = 3,
2. Horse -> Hash function (length) = 5,
3. Elephant -> Hash function (length) = 8,
4. Wolf -> Hash function (length) = 4,
5. Dog -> Hash function (length) = 3.

Taken => look for next free slot (next 2 are also taken!):

Index	Hash table
0	
1	
2	
3	Cat
4	Wolf
5	Horse
6	Dog
7	
8	Elephant

5. LinkedLists (30pts):

a. Answer: See Exercise5.java

6. BONUS (+15pts): Write an algorithm in Java to convert an infix notation into postfix notation:

a. Answer: See Exercise6.java

- i. Source: <https://algorithms.tutorialhorizon.com/convert-infix-to-postfix-expression/>