

"On my honor, I have neither given nor received unauthorized aid in doing this assignment."

PART I.

Summary of learning experience:

The easiest part of the task is to write the code using a *item* struct and an array of type *item*.

The most difficult parts of the task to consider the round-up situation.

The assignment's educational objective: learn how to implement hash map using open addressing technique.

How well I think I achieved them: I think I am doing good for the part 1. I successfully implemented all the functions, and tested them as well.

PART II.

Summary of learning experience:

The easiest part of the task is that most of the functions/ideas are the same as the previous part, so that I only need to change a little bit in this part.

The most difficult parts of the task for me is to insert the items to the list.

The assignment's educational objective is to learn how to implement hash map using open addressing technique.

How well I think I achieved them: I think I am doing good for the part 1. I successfully implemented all the functions, and tested them as well.

PART III.

Summary of learning experience:

The easiest part of the task is that most of the functions/ideas are the same as the previous part. I just add some new hashing functions and change the return types.

The most difficult parts of the task for me is to implement for the hashmap with buckets, because I don't understand why does it needs to probing.

The assignment's educational objective is to learn how to implement hash map using open addressing with different probing methods.

How well I think I achieved them: I think I am not doing good for the hashing with bucket part.

PART IV.

Summary of learning experience:

The easiest part of the task is implement size(), load() functions which were the same as previous part.

The most difficult parts of the task for me is to implement insertion_at_root() and remove() function. We need to consider three cases for the remove() function.

The assignment's educational objective is to learn how to implement randomized binary search tree.

How well I think I achieved them: I think I did not do well for this part. The time is too tight. So many tiny things need to be considered.