# Assignment 3.4

Write a report to demonstrate and explain at least 5 main functions in the unordered\_set. Propose at

least 3 Hackerrank/Codeforces/Leetcode/SPOJ exercises that can be solved by using set data structure.

The `unordered\_set` is an implementation of a set data structure in C++. It is a container that stores unique elements in no particular order. The `unordered\_set` is implemented using hash tables, which makes it very efficient in terms of search, insertion, and deletion operations. In this report, we will demonstrate and explain at least 5 main functions in the `unordered\_set` data structure. We will also propose at least 3 Hackerrank/Codeforces/Leetcode/SPOJ exercises that can be solved using the `set` data structure.

## 1. Insert Function:

The `insert` function is used to insert elements into an `unordered\_set`. It takes a parameter of the element to be inserted. If the element already exists in the set, then the `insert` function does nothing. Otherwise, the element is added to the set.

```

unordered\_set<int> mySet;

mySet.insert(10);

mySet.insert(20);

mySet.insert(30);

```

## 2. Erase Function:

The `erase` function is used to remove elements from an `unordered\_set`. It takes a parameter of the element to be removed. If the element does not exist in the set, then the `erase` function does nothing. Otherwise, the element is removed from the set.

```

unordered\_set<int> mySet;

mySet.insert(10);

mySet.insert(20);

mySet.insert(30);

mySet.erase(20);

```

## 3. Size Function:

The `size` function is used to get the number of elements in an `unordered\_set`. It takes no parameters and returns an integer value representing the number of elements in the set.

```

unordered\_set<int> mySet;

mySet.insert(10);

mySet.insert(20);

mySet.insert(30);

int setSize = mySet.size(); // setSize is 3

```

## 4. Find Function:

The `find` function is used to search for an element in an `unordered\_set`. It takes a parameter of the element to be searched for. If the element exists in the set, then the `find` function returns an iterator pointing to the element. Otherwise, it returns an iterator pointing to the end of the set.

```

unordered\_set<int> mySet;

mySet.insert(10);

mySet.insert(20);

mySet.insert(30);

auto it = mySet.find(20); // it points to 20

auto it2 = mySet.find(40); // it2 points to mySet.end()

```

## 5. Clear Function:

The `clear` function is used to remove all elements from an `unordered\_set`. It takes no parameters and returns nothing.

```

unordered\_set<int> mySet;

mySet.insert(10);

mySet.insert(20);

mySet.insert(30);

mySet.clear(); // mySet is now empty

```

Hackerrank/Codeforces/Leetcode/SPOJ exercises that can be solved using `unordered\_set`:

## 1. Leetcode - Two Sum: <https://leetcode.com/problems/two-sum/>

## 2. Hackerrank - Pairs: <https://www.hackerrank.com/challenges/pairs/problem>

## 3. Codeforces - Two Substrings: <https://codeforces.com/problemset/problem/550/A>