

Module sui::vec_set

A set data structure backed by a vector. The set is guaranteed not to contain duplicate keys. All operations are $O(N)$ in the size of the set

This key already exists in the map

This key does not exist in the map

Create an empty [VecSet](#)

Create a singleton [VecSet](#) that only contains one element.

Insert a key into self. Aborts if key is already present in self.

Remove the entry key from self. Aborts if key is not present in self.

Return true if self contains an entry for key, false otherwise

Return the number of entries in self

Return true if self has 0 elements, false otherwise

Unpack self into vectors of keys. The output keys are stored in insertion order, not sorted.

Construct a new [VecSet](#) from a vector of keys. The keys are stored in insertion order (the original [keys](#) ordering) and are not sorted.

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not sorted.

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

Struct

A set data structure backed by a vector. The set is guaranteed not to contain duplicate keys. All operations are $O(N)$ in the size of the set

```
```bash
```

```
```
```

This key already exists in the map

```
```bash
```

```
```
```

This key does not exist in the map

```
```bash
```

```
```
```

Create an empty [VecSet](#)

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Create a singleton [VecSet](#) that only contains one element.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Insert a key into self. Aborts if key is already present in self.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Remove the entry key from self. Aborts if key is not present in self.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self contains an entry for key, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return the number of entries in self

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self has 0 elements, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Unpack self into vectors of keys. The output keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Construct a new [VecSet](#) from a vector of keys. The keys are stored in insertion order (the original [keys](#) ordering) and are not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Constants

This key already exists in the map

```
'''bash
```

```
'''
```

This key does not exist in the map

```
'''bash
```

```
'''
```

Create an empty [VecSet](#)

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Create a singleton [VecSet](#) that only contains one element.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Insert a key into self. Aborts if key is already present in self.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Remove the entry key from self. Aborts if key is not present in self.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self contains an entry for key, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return the number of entries in self

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self has 0 elements, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Unpack self into vectors of keys. The output keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
```bash
```

```
```
```

Construct a new [VecSet](#) from a vector of keys. The keys are stored in insertion order (the original [keys](#) ordering) and are not sorted.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not sorted.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Function

Create an empty [VecSet](#)

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Create a singleton [VecSet](#) that only contains one element.

```
```bash
```

```
```
```

```
```bash
```

```
'''
```

Insert a key into self. Aborts if key is already present in self.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Remove the entry key from self. Aborts if key is not present in self.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self contains an entry for key, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return the number of entries in self

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self has 0 elements, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Unpack self into vectors of keys. The output keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Construct a new [VecSet](#) from a vector of keys. The keys are stored in insertion order (the original [keys](#) ordering) and are not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Create a singleton [VecSet](#) that only contains one element.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Insert a key into self. Aborts if key is already present in self.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Remove the entry key from self. Aborts if key is not present in self.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self contains an entry for key, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return the number of entries in self

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self has 0 elements, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Unpack self into vectors of keys. The output keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Construct a new [VecSet](#) from a vector of keys. The keys are stored in insertion order (the original [keys](#) ordering) and are not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
'''bash
```



```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Insert a key into self. Aborts if key is already present in self.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Remove the entry key from self. Aborts if key is not present in self.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self contains an entry for key, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return the number of entries in self

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self has 0 elements, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Unpack self into vectors of keys. The output keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Construct a new [VecSet](#) from a vector of keys. The keys are stored in insertion order (the original [keys](#) ordering) and are not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Remove the entry key from self. Aborts if key is not present in self.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self contains an entry for key, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return the number of entries in self

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self has 0 elements, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Unpack self into vectors of keys. The output keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Construct a new [VecSet](#) from a vector of keys. The keys are stored in insertion order (the original [keys](#) ordering) and are not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Return true if self contains an entry for key, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return the number of entries in self

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self has 0 elements, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Unpack self into vectors of keys. The output keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Construct a new [VecSet](#) from a vector of keys. The keys are stored in insertion order (the original [keys](#) ordering) and are not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not

sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Return the number of entries in self

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Return true if self has 0 elements, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Unpack self into vectors of keys. The output keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Construct a new [VecSet](#) from a vector of keys. The keys are stored in insertion order (the original [keys](#) ordering) and are not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Return true if self has 0 elements, false otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Unpack self into vectors of keys. The output keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Construct a new [VecSet](#) from a vector of keys. The keys are stored in insertion order (the original [keys](#) ordering) and are not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Unpack self into vectors of keys. The output keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Construct a new [VecSet](#) from a vector of keys. The keys are stored in insertion order (the original [keys](#) ordering) and are not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Construct a new [VecSet](#) from a vector of keys. The keys are stored in insertion order (the original [keys](#) ordering) and are not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```



## Function

Borrow the contents of the [VecSet](#) to access content by index without unpacking. The contents are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Find the index of key in self. Return None if key is not in self. Note that keys are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Find the index of key in self. Aborts if key is not in self. Note that map entries are stored in insertion order, not sorted.

```
'''bash
```

```
'''
```

```
'''bash
```

