Table of Contents

HashMap

A mutable hash map based on a Robin Hood hash table.

Usage

Create

You can create an empty map using new() or construct it using from_array().

```

test {
  let _map2 : @hashmap.HashMap[String, Int] = @hashmap.new()

}
```

Set & Get

You can use set() to add a key-value pair to the map, and use get() to get a value.

```

test {
  let map : @hashmap.HashMap[String, Int] = @hashmap.new()
  map.set("a", 1)
  assert_eq(map.get("a"), Some(1))
  assert_eq(map.get_or_default("a", 0), 1)
  assert_eq(map.get_or_default("b", 0), 0)
  map.remove("a")
  assert_eq(map.contains("a"), false)
}
```

Remove

You can use remove() to remove a key-value pair.

```

test {
  let map = @hashmap.of([("a", 1), ("b", 2), ("c", 3)])
  map.remove("a") |> ignore
  assert_eq(map.to_array(), [("c", 3), ("b", 2)])
}
```

Contains

You can use contains() to check whether a key exists.

```

test {
  let map = @hashmap.of([("a", 1), ("b", 2), ("c", 3)])
  assert_eq(map.contains("a"), true)
  assert_eq(map.contains("d"), false)
}
```

Size & Capacity

You can use size() to get the number of key-value pairs in the map, or capacity() to get the current capacity.

```

test {
  let map = @hashmap.of([("a", 1), ("b", 2), ("c", 3)])
  assert_eq(map.size(), 3)
  assert_eq(map.capacity(), 8)
}
```

Similarly, you can use is_empty() to check whether the map is empty.

```

test {
  let map : @hashmap.HashMap[String, Int] = @hashmap.new()
  assert_eq(map.is_empty(), true)
}
```

Clear

You can use clear to remove all key-value pairs from the map, but the allocated memory will not change.

```

test {
  let map = @hashmap.of([("a", 1), ("b", 2), ("c", 3)])
  map.clear()
  assert_eq(map.is_empty(), true)
}
```

Iteration

You can use each() or eachi() to iterate through all key-value pairs.

```

test {
  let map = @hashmap.of([("a", 1), ("b", 2), ("c", 3)])
  let arr = []
  map.each((k, v) => arr.push((k, v)))
  let arr2 = []
  map.eachi((i, k, v) => arr2.push((i, k, v)))
}
```

Or use iter() to get an iterator of hashmap.

```

test {
  let map = @hashmap.of([("a", 1), ("b", 2), ("c", 3)])
  let _iter = map.iter()

}
```