Data Structures

CSCI-UA-102-011-Spring-2025

Recitation - 8

- Rasmika Billa
(03/14/2025)

Note

- Section 3 of the assignment is removed!
- You are allowed 1 sheet of handwritten note for the Quiz (double-sided is also fine)

Agenda

- Q10.12
- Q10.6
- Q10.10

10.12

1 (Initialization)

```
import java.util.Objects;

public class Pair<A, B> {
   private final A first;
   private final B second;

   public Pair(A a, B b) {
      this.first = a;
      this.second = b;
   }

   public A getFirst() {
      return first;
   }

   public B getSecond() {
      return second;
   }
```

2 (Check Pairs)

```
 @Override
   public boolean equals(Object obj) {
      if (this == obj) return true;
      if (obj == null || getClass() != obj.getClass()) return false;
      Pair<?, ?> pair = (Pair<?, ?>) obj;
      return Objects.equals(first, pair.first) &&
Objects.equals(second, pair.second);
   }
```

3 (Hash Function)

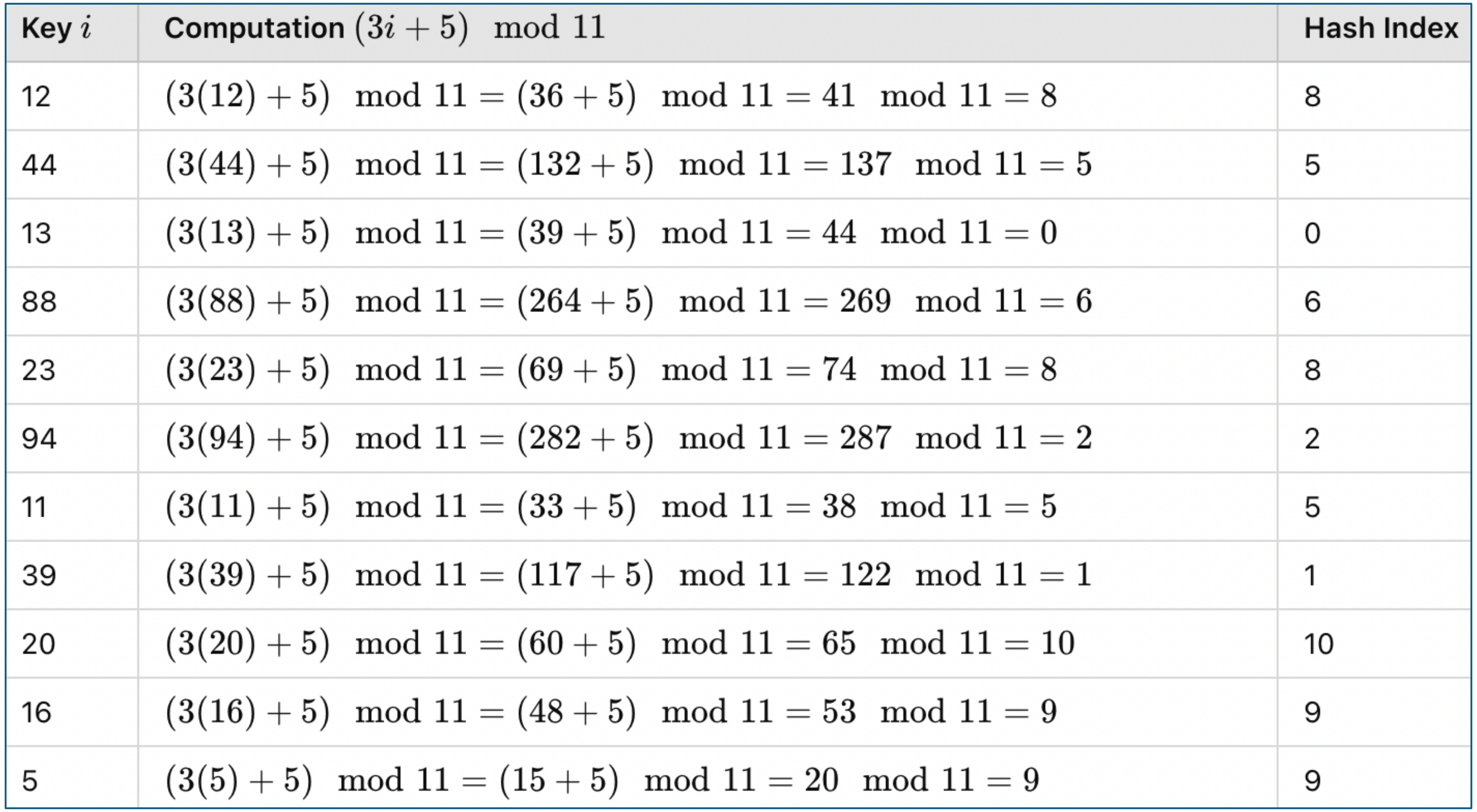
```
@Override
   public int hashCode() {
      return Objects.hash(first, second);
   }

   @Override
   public String toString() {
      return "Pair{" + "first=" + first + ", second=" + second + '}';
   }
```

hash = 31 * hash + charAt(i);
(default)

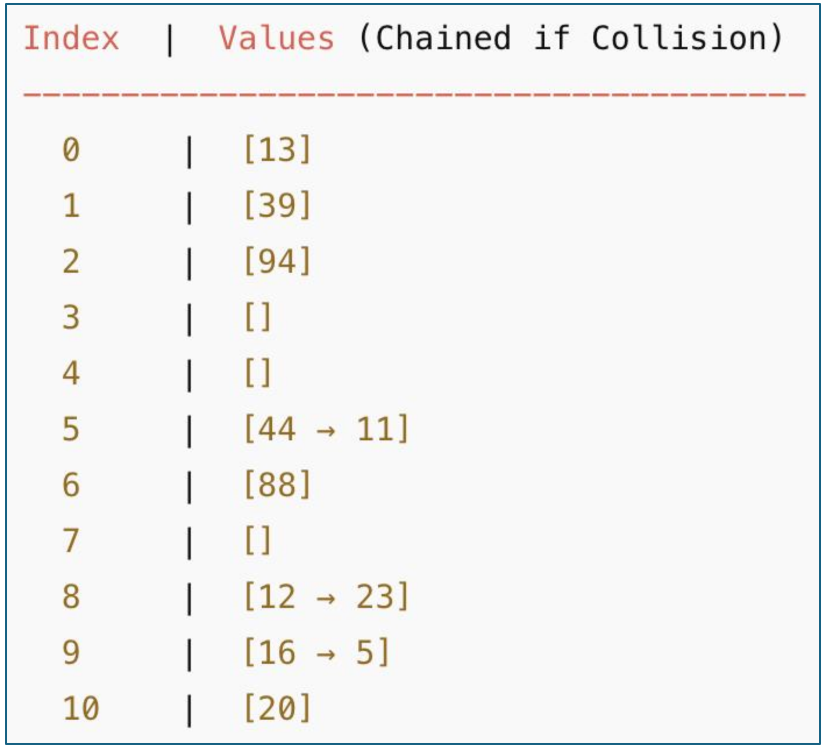
4 (Main method)

```
 public static void main(String[] args) {
      Pair<String, Integer> pair1 = new Pair<>("Hello", 42);
      Pair<String, Integer> pair2 = new Pair<>("Hello", 42);
      Pair<String, Integer> pair3 = new Pair<>("World", 99);
      System.out.println(pair1);
      System.out.println(pair1.equals(pair2));
      System.out.println(pair1.equals(pair3));
      System.out.println(pair1.hashCode());
}
```

10.6



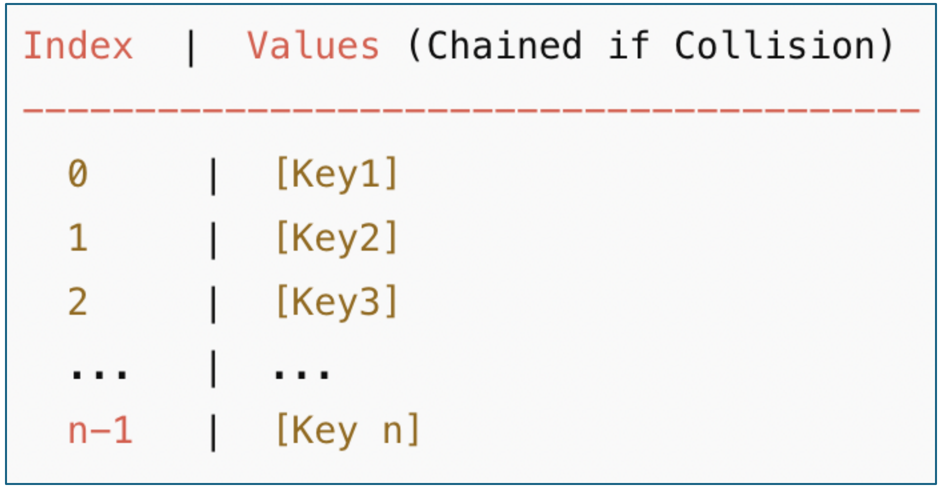
Key i	Computation $(3i+5) \mod 11$	Hash Index
12	$(3(12)+5) \mod 11 = (36+5) \mod 11 = 41 \mod 11 = 8$	8
44	$(3(44)+5) \mod 11 = (132+5) \mod 11 = 137 \mod 11 = 5$	5
13	$(3(13)+5) \mod 11 = (39+5) \mod 11 = 44 \mod 11 = 0$	0
88	$(3(88)+5) \mod 11 = (264+5) \mod 11 = 269 \mod 11 = 6$	6
23	$(3(23)+5) \mod 11 = (69+5) \mod 11 = 74 \mod 11 = 8$	8
94	$(3(94)+5) \mod 11 = (282+5) \mod 11 = 287 \mod 11 = 2$	2
11	$(3(11)+5) \mod 11 = (33+5) \mod 11 = 38 \mod 11 = 5$	5
39	$(3(39)+5) \mod 11 = (117+5) \mod 11 = 122 \mod 11 = 1$	1
20	$(3(20)+5) \mod 11 = (60+5) \mod 11 = 65 \mod 11 = 10$	10
16	$(3(16)+5) \mod 11 = (48+5) \mod 11 = 53 \mod 11 = 9$	9
5	$(3(5)+5) \mod 11 = (15+5) \mod 11 = 20 \mod 11 = 9$	9



```
Index    |  Values (Chained if Collision)
-----------------------------------------

  0      |  [13]
  1      |  [39]
  2      |  [94]
  3      |  []
  4      |  []
  5      |  [44 → 11]
  6      |  [88]
  7      |  []
  8      |  [12 → 23]
  9      |  [16 → 5]
  10     |  [20]
```

How many collisions do we have? And how did we handle that??

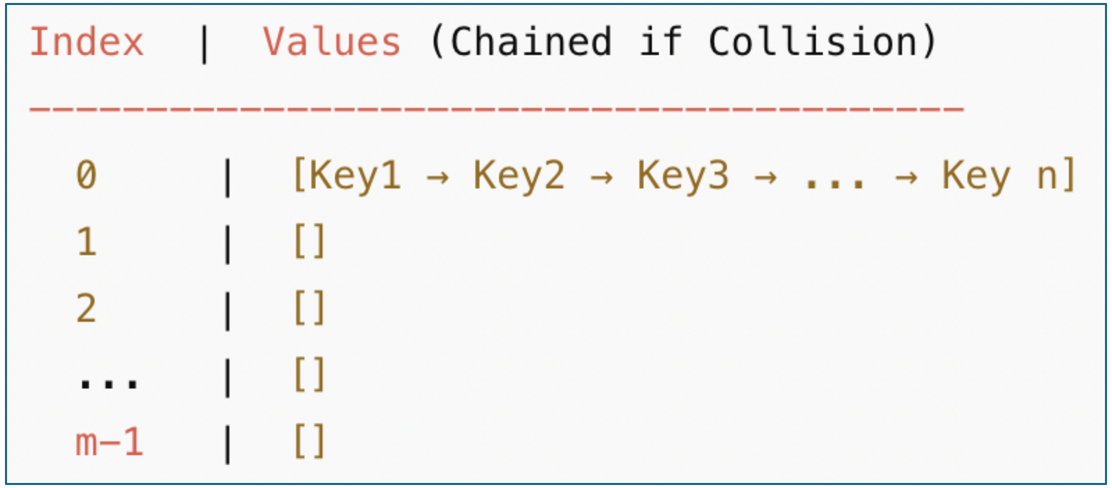
10.10



```
Index   |  Values (Chained if Collision)
----------------------------------------
  0     |  [Key1]
  1     |  [Key2]
  2     |  [Key3]
  ...   |  ...
  n−1   |  [Key n]
```

O(n)

- Each insertion takes **constant time** O(1) since each value is directly placed in an empty bucket.
- **Total time for n insertions:** O(n).



```
Index   |  Values (Chained if Collision)
----------------------------------------
  0     |  [Key1 → Key2 → Key3 → ... → Key n]
  1     |  []
  2     |  []
  ...   |  []
  m−1   |  []
```

O(n^2)

- Each insertion involves **traversing an entire linked list** of existing elements, leading to increasing cost as more elements are added.
- **Total time for n insertions:** O(n^2).