# 整数集合

整数集合（ intset ）是集合键的底层实现之一，当一个集合只包含整数值元素，并且这个集合的元素数量不多时， Redis 就会使用整数集合作为集合键的底层实现。

## 整数集合的实现

整数集合（ intset ）是Redis 用于保存整数值的集合抽象数据结构，它可以保存类型为int16\_t 、int32\_t 或者int64\_t 的整数值，并且保证集合中不会出现重复元素。每个intset.h/intset 结构表示一个整数集合：

|  |
| --- |
| typedef struct intset {    // 编码方式  uint32\_t encoding;  // 集合包含的元素数量  uint32\_t length;  // 保存元素的数组  int8\_t contents[];  } intset; |

contents 数组是整数集合的底层实现：整数集合的每个元素都是contents 数组的一个数组项（item），各个项在数组中按值的大小从小到大有序地排列，并且数组中不包含任何重复项。

length 属性记录了整数集合包含的元素数量，也即是contents 数组的长度。虽然intset 结构将contents 属性声明为int8\_t 类型的数组，但实际上contents 数组并不保存任何int8\_t 类型的值， contents 数组的真正类型取决于encoding 属性的值：

* 如果encoding 属性的值为INTSET\_ENC\_INT16 ，那么contents 就是一个int16\_t 类型的数组，数组里的每个项都是一个int16 t 类型的整数值（最小值为-32768，最大值为32767 ）。
* 如果encoding 属性的值为工NTSET\_ENC\_INT32 ，那么contents 就是一个int32\_t 类型的数组，数组里的每个项都是一个int32 t 类型的整数值（最小值为-2147483648 ，最大值为2147483647 ）。
* 如果encoding 属性的值为工NTSET\_ENC\_INT64 ，那么contents 就是一个int64\_t 类型的数组，数组里的每个项都是一个int64 t 类型的整数值（最小值为-9 223372036854775808 ，最大值为9223372036854775807）。

## 升级

每当我们要将一个新元素添加到整数集合里面，并且新元素的类型比整数集合现有所有元素的类型都要长时，整数集合需要先进行升级（ upgrade ），然后才能将新元素添加到整数集合里面。

升级整数集合并添加新元素共分为三步进行：

1. 根据新元素的类型，扩展整数集合底层数组的空间大小，并为新元素分配空间。
2. 将底层数组现有的所有元素都转换成与新元素相同的类型，并将类型转换后的元素放置到正确的位上，而且在放置元素的过程中，需要继续维持底层数组的有序性质不变。
3. 将新元素添加到底层数组里面。

**升级之后新元素的摆踵位置**

因为引发升级的新元素的长度总是比整数集合现有所有元素的长皮都大，所以这个新元素的值要么就大于所有现有元素，要么就小于所有现有元素：

* 在新元素小于所有现有元素的情况下，新元素会被放置在底层数组的最开头（牵引0);
* 在新元，素大于所有现有元素的情况下，新元，素会被放直在底层数组的最末尾（索引length-1）。

## 升级的好处

整数集合的升级策略有两个好处，一个是提升整数集合的灵活性，另一个是尽可能地节

## 降级

整数集合不支持降级操作，一旦对数组进行了升级，编码就会一直保持升级后的状态。

## 整数集合API

表6-1 列出了整数集合的操作API 。

