

牛客大学高薪加成系列课

# 整数集合



整数集合 (intset) 是一个有序的、存储整型数据的结构：

1. 整数集合中的元素按照值的大小由小到大排列；
2. 它可以保存int16\_t、int32\_t、int64\_t类型的整数值；
3. 在存储数据时，整数集合可以保证其内部不出现重复的数据。

当一个set只包含整数元素，并且这个set的元素数量不多时，Redis就会使用整数集合作为set的底层实现！

```
/* intset.h */  
  
typedef struct intset {  
    uint32_t encoding;    // 编码类型  
    uint32_t length;      // 元素数量  
    int8_t contents[];    // 元素数组  
} intset;
```

`contents`虽然声明为`int8_t`类型，但它实际上并不保存任何`int8_t`类型的值，`contents`数组的实际类型取决于`encoding`属性的值：

1. `encoding = INTSET_ENC_INT16` -> `contents`存储`int16_t`类型的值；
2. `encoding = INTSET_ENC_INT32` -> `contents`存储`int32_t`类型的值；
3. `encoding = INTSET_ENC_INT64` -> `contents`存储`int64_t`类型的值；

`INTSET_ENC_INT16`、`INTSET_ENC_INT32`、`INTSET_ENC_INT64`定义在`intset.c`中。

### ■ 升级

当添加的新元素，其类型比现有元素类型都长时，集合需要先升级再添加：

1. 根据新元素的类型，扩展集合底层数组的空间，并为新元素分配空间；
2. 将现有元素都转成与新元素相同的类型，并将其存储到正确的位置上；
3. 将新元素添加到数组之内；

### ■ 降级

整数集合不支持降级，一旦对数组进行了升级，编码就会一直保持升级后的状态！

升级的优点	让一个整数数组同时支持 <code>int16_t</code> 、 <code>int32_t</code> 、 <code>int64_t</code> ，最简单的做法是使用 <code>int64_t</code> ，但这样显然是浪费内存空间的，而升级操作可以尽可能的节约内存的使用。
升级的缺点	每次向集合中添加数据都可能会引起升级，而每次升级都需要对底层数组中所有的元素进行类型转换，所以向集合中添加新元素的时间复杂度为 $O(n)$ 。



# 牛客大学

- 专业求职辅导 -

# THANKS



关注【牛客大学】公众号  
回复“牛客大学”获取更多求职资料