如何对Scala中集合(Collections)进行排序

下面是一系列对Scala中的Lists、Array进行排序的例子，数据结构的定义如下：

```
// data structures working with
val s = List( "a", "d", "F", "B", "e")
val n = List(3, 7, 2, 1, 5)
val m = Map(
   -2 -> 5,
   2 -> 6,
   5 -> 9,
   1 -> 2,
   0 -> -16,
   -1 -> -4
)
```

利用Scala内置的sorted方法进行排序

```
s.sorted
res0: List = List(B, F, a, d, e)

n.sorted
res1: List[Int] = List(1, 2, 3, 5, 7)
```

当然，我们也可以自定义排序规则：

```
implicit val KeyOrdering = new Ordering[String] {
     override def compare(x: String, y: String): Int = {
       y.compareTo(x)
     }
}
s.sorted
List(e, d, a, F, B)
```

为什么我们这里不对m也排序呢？这是因为map对象没有sorted方法！

大小写敏感搜索

我们可以用Scala中的sortWith来自定义我们的对大小写敏感的排序函数。代码如下：

```
/* sort alphabetical and ignoring case */
def compfn1(e1: String, e2: String) = (e1 compareToIgnoreCase e2) < 0

/* sort alphabetical and ignoring case: alternate */
def compfn2(e1: String, e2: String) = (e1.toLowerCase < e2.toLowerCase)

s.sortWith(compfn1)
res2: List = List(a, B, d, e, F)

s.sortWith(compfn2)
res3: List = List(a, B, d, e, F)

/* Or you can do so using anonymous function (Thanks Rahul) */
s.sortWith(_.toLowerCase < _.toLowerCase)
res4: List = List(a, B, d, e, F)
```

如何对Map中的Key或Value进行排序

其实很简单代码如下：

```
// sort by key can use sorted
m.toList.sorted foreach {
   case (key, value) =>
      println(key + " = " + value)
}

-2 = 5
-1 = -4
0 = -16
1 = 2
2 = 6
5 = 9

// sort by value
m.toList sortBy ( _._2 ) foreach {
   case (key, value) =>
```

```
    println(key + " = " + value)
}

0 = -16
-1 = -4
1 = 2
-2 = 5
2 = 6
5 = 9
```

对源数据排序

上面的排序并不对原始的数据产生影响，排序的结果被存储到别的变量中，如果你的元素类型是数组，那么你还可以对数组本身进行排序，如下：

```
scala> val a = Array(2,6,1,9,3,2,1,-23)
a: Array[Int] = Array(2, 6, 1, 9, 3, 2, 1, -23)

scala> scala.util.Sorting.quickSort(a)

scala> a.mkString(",")
res24: String = -23,1,1,2,2,3,6,9
```

可以看到a数组内部的数据已经排好序了。
如果你对上面的n进行排序，发现会报出如下的错误：

```
scala> scala.util.Sorting.quickSort(n)
<console>:14: error: overloaded method value quickSort with alternatives:
  (a: Array[Float])Unit <and>
  (a: Array[Int])Unit <and>
  [K](a: Array[K])(implicit evidence$1: scala.math.Ordering[K])Unit <and>
  (a: Array[Double])Unit
 cannot be applied to (List[Int])
           scala.util.Sorting.quickSort(n)
```

从上面的报错信息我们可以看出，只有Array才可以用scala.util.Sorting.quickSort方法。
在scala.util.Sorting下面还有个stableSort函数，它可以对所有Seq进行排序，返回的结果为A

rray。比如我们对上面的n进行排序：

```
scala> scala.util.Sorting.stableSort(n)
res35: Array[Int] = Array(1, 2, 3, 5, 7)
```

而对Array排序返回Unit

```
scala> val a = Array(2,6,1,9,3,2,1,-23)
a: Array[Int] = Array(2, 6, 1, 9, 3, 2, 1, -23)

scala> scala.util.Sorting.stableSort(a)

scala> a.mkString(",")
res39: String = -23,1,1,2,2,3,6,9
```

从名字上我们也可以看出，stableSort是稳定排序。所以可以根据需要进行选择。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（过往记忆）所有，未经许可不得转载。
本文链接:【】（）