

阅读: 128100

排序算法

排序也是在程序中经常用到的算法。无论使用冒泡排序还是快速排序，排序的核心是比较两个元素的大小。如果是数字，我们可以直接比较，但如果是字符串或者两个dict呢？直接比较数学上的大小是没有意义的，因此，比较的过程必须通过函数抽象出来。

Python内置的 `sorted()` 函数就可以对list进行排序：

```
>>> sorted([36, 5, -12, 9, -21])
[-21, -12, 5, 9, 36]
```

此外，`sorted()` 函数也是一个高阶函数，它还可以接收一个 `key` 函数来实现自定义的排序，例如按绝对值大小排序：

```
>>> sorted([36, 5, -12, 9, -21], key=abs)
[5, 9, -12, -21, 36]
```

`key` 指定的函数将作用于list的每一个元素上，并根据`key`函数返回的结果进行排序。对比原始的list和经过 `key=abs` 处理过的list：

```
list = [36, 5, -12, 9, -21]

keys = [36, 5, 12, 9, 21]
```

然后 `sorted()` 函数按照keys进行排序，并按照对应关系返回list相应的元素：

```
keys排序结果 => [5, 9, 12, 21, 36]
               | |   |   |   |
最终结果      => [5, 9, -12, -21, 36]
```

我们再看一个字符串排序的例子：

```
>>> sorted(['bob', 'about', 'Zoo', 'Credit'])
['Credit', 'Zoo', 'about', 'bob']
```

默认情况下，对字符串排序，是按照ASCII的大小比较的，由于 `'Z' < 'a'`，结果，大写字母 `Z` 会排在小写字母 `a` 的前面。

现在，我们提出排序应该忽略大小写，按照字母序排序。要实现这个算法，不必对现有代码大加改动，只要我们能用一个`key`函数把字符串映射为忽略大小写排序即可。忽略大小写来比较两个字符串，实际上就是先把字符串都变成大写（或者都变成小写），再比较。

这样，我们给 `sorted` 传入`key`函数，即可实现忽略大小写的排序：

```
>>> sorted(['bob', 'about', 'Zoo', 'Credit'], key=str.lower)
['about', 'bob', 'Credit', 'Zoo']
```

要进行反向排序，不必改动`key`函数，可以传入第三个参数`reverse=True`：

```
>>> sorted(['bob', 'about', 'Zoo', 'Credit'], key=str.lower, reverse=True)
['Zoo', 'Credit', 'bob', 'about']
```

从上述例子可以看出，高阶函数的抽象能力是非常强大的，而且，核心代码可以保持得非常简洁。

小结

`sorted()` 也是一个高阶函数。用 `sorted()` 排序的关键在于实现一个映射函数。

练习

假设我们用一组`tuple`表示学生名字和成绩：

```
L = [('Bob', 75), ('Adam', 92), ('Bart', 66), ('Lisa', 88)]
```

请用 `sorted()` 对上述列表分别按名字排序：

```
# -*- coding: utf-8 -*-
```

```
L = [('Bob', 75), ('Adam', 92), ('Bart', 66), ('Lisa', 88)]
```

```
def by_name(t):
```

```
    pass
```

```
L2 = sorted(L, key=by_name)
print(L2)
```

▶ Run

再按成绩从高到低排序：

```
# -*- coding: utf-8 -*-
```

```
L = [('Bob', 75), ('Adam', 92), ('Bart', 66), ('Lisa', 88)]
```

```
def by_score(t):
    pass
```

L2 = ???

```
print(L2)
```

▶ Run

参考源码

[do_sorted.py](#)

感觉本站内容不错，读后有收获？

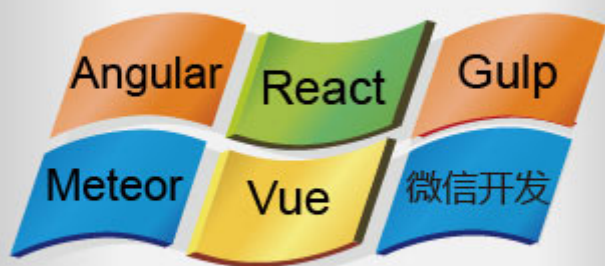
¥ 我要小额赞助，鼓励作者写出更好的教程

还可以分享给朋友

➤ 分享  clind  2人分享过



珠峰NODE.JS全栈开发



技术陪伴成长社区

 珠峰培训

 麦子学院
www.maiziedu.com

百万级python导师亲身指导

保你120天 变身python大牛

有时候，你需要的只是一句点拨

立即咨询

 掘金

一个只有高手分享
的技术社区

深度学习在线课程

通向无人驾驶的必经之路



[← filter](#)

[返回函数](#) [➤](#)

评论

发表评论

Sign In to Make a Comment



友情链接: [中华诗词](#) - [阿里云](#) - [SICP](#) - [4closure](#)