

DSHomework3

Find Based Frequency

设计思路

- 因为需要查找对象并进行更新，因而用List存储所有的对象
- 由于需要记录对象的频度，且需要根据频度进行调整，因而采用Map存储每个对象的访问频度
- 因为序列是基于频度排序的，因而插入对象时需要将对象的初始频度设置为0
- 在调整序列时，可以记录查找对象前一个的索引，然后向前遍历，直至大于新频度时，然后在此处插入原对象
- 因对象需要删除和插入，故设置newIndex以表征新插入位置，index表征删除位置
- 对于频度一致的对象，基于查找顺序，先达到同一频度的对象排在序列的前面

实现代码

具体实现代码如下：

```
1
2 public class FrequencyBasedLookup {
3     private Map<String, Integer> frequencyMap;//存储频度
4     private List<String> lookupTable;//存储对象
5
6     public FrequencyBasedLookup() {
7         frequencyMap = new HashMap<>();
8         lookupTable = new ArrayList<>();
9     }
10
11     public void addObject(String object) {
12         lookupTable.add(object);
13         frequencyMap.put(object, 0);
14     }
15
16     public int searchObject(String object) {
17         int index = lookupTable.indexOf(object);//查找对象的索引
18         if (index != -1) {
19             int frequency = frequencyMap.get(object) + 1;
20             frequencyMap.put(object, frequency);
21             int newIndex = index - 1;
22             while (newIndex >= 0 &&
23 frequencyMap.get(lookupTable.get(newIndex)) < frequency)
24                 newIndex--;
25         }
26         newIndex++;//找到插入位置
27         if (newIndex != index) {
28             lookupTable.remove(index);//删除原先结点
29             lookupTable.add(newIndex, object);//插入新结点
30         }
```

```
31         }
32         return index;
33     }
34 }
35
```

优化思路

- 可以使用Tree来存储频度，以便更快地进行查找和替换操作