

Algorithms – Programming Assignment 3

Autocomplete

B06602035 李晴妍

E-Mail : b06602035@ntu.edu.tw

Phone : 0987883537

1.
 - (1) Operating system: Windows
 - (2) Compiler: IntelliJ IDEA
 - (3) Text editor / IDE: IntelliJ IDEA
2. Describe how your `firstIndexOf()` method in `BinarySearchDeluxe.java` finds the first index of a key that is equal to the search key.

一開始要先檢查所有 argument 不可以等於 `null`。

實現的方式是採用 `binary search`，因為傳入的 array `a` 已經 `sort` 過了，所以比較 `key` 與 `lo` 和 `hi` 正中間 `mid` 的大小。如果 `key` 比 `mid` 大，把 `lo` 改成現在 `mid` 的右邊一個，繼續比較右半邊。那如果 `key` 比 `mid` 小，就用相同方式往左半邊找。如果 `key` 與 `mid` 相同，則不斷把 `mid-1`，直到 `mid < 0` 或 `mid` 與 `key` 不同。就可以找到第一個 `key` 的 index。

```
15 // Returns the index of the first key in the sorted array a[]
16 // that is equal to the search key, or -1 if no such key.
17 public static <Key> int firstIndexOf(Key[] a, Key key, Comparator<Key> comparator) {
18     if (a == null || key == null || comparator == null) throw new IllegalArgumentException();
19
20     int lo = 0;
21     int hi = a.length - 1;
22
23     while (hi >= lo) {
24         int mid = lo + (hi - lo) / 2;
25         if (comparator.compare(key, a[mid]) > 0) lo = mid + 1;
26         else if (comparator.compare(key, a[mid]) < 0) hi = mid - 1;
27         else {
28             while (mid >= 0 && comparator.compare(a[mid], key) == 0)
29                 mid--;
30             return mid + 1;
31         }
32     }
33
34     return -1;
35 }
36
```

3. Identify which sorting algorithm (if any) that your program uses in the Autocomplete constructor and instance methods.
If you are using an optimized implementation, such as `Arrays.sort()`, select the principal algorithm.

`Autocomplete()` : `Arrays.sort()`

`allMatches()` : `Arrays.sort()`

`numberOfMatches()` : `Arrays.sort()`

4. How many compares (in the worst case) does each of the operations in the Autocomplete data type make, as a function of both the number of terms `n`

and the number of matching terms m ? Use Big Theta notation to simplify your answers.

Autocomplete(): $\Theta(n \log n)$

Default constructor 裡有重頭檢查 $\text{array}(n)$ · 最後有用 $\text{Arrays.sort}() (n \log n)$ · 所以 $\Theta(n + n \log n) \sim \Theta(n \log n)$ ·

```
16 // Initializes the data structure from the given array of terms.
17 public Autocomplete(Term[] terms) {
18     if (terms == null) throw new java.lang.IllegalArgumentException();
19     for (Term i : terms) {
20         if (i == null) throw new java.lang.IllegalArgumentException();
21     }
22
23     this.terms = terms;
24     Arrays.sort(this.terms);
25 }
26
```

allMatches(): $\Theta(n + m \log m)$

裡面包含 $\text{firstIndexOf}()$ 及 $\text{lastIndexOf}()$ · 還有 k matches · 以及最後的 $\text{Array.sort}()$ · 所以在 worst case · $k = n$ · $\Theta(2 \cdot 0.5n + k + m \log m)$
 $\sim \Theta(n + n + n \log n) \sim \Theta(n + m \log m)$

```
27 // Returns all terms that start with the given prefix, in descending order of weight.
28 public Term[] allMatches(String prefix) {
29     if (prefix == null) throw new NullPointerException();
30     Term a = new Term(prefix, 0);
31
32     int start = BinarySearchDeLuxe.firstIndexOf(terms, a, Term.byPrefixOrder(prefix.length()));
33     if (start < 0) return new Term[] { };
34     int end = BinarySearchDeLuxe.lastIndexOf(terms, a, Term.byPrefixOrder(prefix.length()));
35     if (end < 0) return new Term[] { };
36
37     Term[] b = new Term[end - start + 1];
38     for (int i = start; i <= end; i++) {
39         b[i - start] = terms[i];
40     }
41
42     Arrays.sort(b, Term.byReverseWeightOrder());
43     return b;
44 }
```

numberOfMatches(): $\Theta(n)$

是由 $\text{firstIndexOf}()$ 跟 $\text{lastIndexOf}()$ 實作 · 所以在 worst case · $\Theta(2 \cdot 0.5n)$
 $\sim \Theta(n)$ ·

```
46 // Returns the number of terms that start with the given prefix.
47 public int numberOfMatches(String prefix) {
48     if (prefix == null) throw new NullPointerException();
49     Term a = new Term(prefix, 0);
50
51     int start = BinarySearchDeLuxe.firstIndexOf(terms, a, Term.byPrefixOrder(prefix.length()));
52     if (start < 0) return 0;
53     int end = BinarySearchDeLuxe.lastIndexOf(terms, a, Term.byPrefixOrder(prefix.length()));
54     if (end < 0) return 0;
55
56     return end - start + 1;
57 }
58
```

5. Describe whatever help (if any) that you received. Don't include readings, lectures, and precepts, but do include any help from people (including course staff, lab TAs, classmates, and friends) and attribute them by name.

(1) https://github.com/ritsu/Algorithms/blob/master/src/main/java/net/qiguang/algorithms/P3_Autocomplete/Autocomplete.java

(2) 也有跟同學討論！

6. Describe any serious problems you encountered.

(1) 在最後跑 GUI 的時候，一直有錯跑不出來就放了幾天沒管。後來寫 report 的時候突然發現是因為其他幾個 java 檔的 main 沒有註解掉，所以 GUI 的 main 才有 error。

(2) 在 Autocomplete.java 裡的 allMatches() 和 numberOfMatches()，如果有回傳值就不需要重複執行。