

16 . [Java] ??? ??? - sb.charAt()

<https://school.programmers.co.kr/learn/courses/30/lessons/181913>

??

- `my_string` 2 ≤ 길이 ≤ 10,000, `queries` 길이 ≤ 10,000
- `queries`의 각 `[s, e]`는 `my_string`의 `s` ≤ `e`를 만족하는 범위이다
- 각 쿼리마다 `my_string[s:e]`의 문자를 정렬하여 반환한다

????

- `my_string`: 길이 1~1,000
- `queries`: 길이 1~1,000
 - `0 ≤ s ≤ e < my_string.length()`

??

1. `StringBuilder`를 사용하여 문자를 정렬한다
2. `[s, e]` 범위의 문자를 추출한다
3. 추출한 문자를 정렬하여 반환한다

```
import java.util.*;

class Solution {
    public String solution(String s) {
        int[] cnt = new int[26];

        for(char c : s.toCharArray()) {
            cnt[c - 'a']++;
        }

        StringBuilder sb = new StringBuilder();
```

```

        for (int i = 0; i < 26; i++) {
            if (cnt[i] == 1) sb.append((char)i+'a');
        }
        return sb.toString();
    }
}

```

1. 문자열을 배열로 변환하고, 각 문자의 출현 횟수를 카운트한다.

```

class Solution {
    public String solution(String my_string, int[][] queries) {
        StringBuilder sb = new StringBuilder(my_string);

        // 각 쿼리마다 문자를 뒤집는다
        for (int[] query : queries) {
            int s = query[0];
            int e = query[1];
            reverse(sb, s, e); // [s, e] 범위 내의 문자를 뒤집는다
        }

        return sb.toString();
    }

    // swap 함수 (swap 함수)
    private void reverse(StringBuilder sb, int start, int end) {
        while (start < end) {
            swap(sb, start, end); // start와 end 위치의 문자를 swap
            start++;
            end--;
        }
    }

    // swap 함수 (swap 함수)
    private void swap(StringBuilder sb, int i, int j) {
        char temp = sb.charAt(i);
        sb.setCharAt(i, sb.charAt(j));
        sb.setCharAt(j, temp);
    }
}

```



swap() 参数是索引 i 和 j，交换 sb 中索引 i 和 j 的字符。

```
private void swap(StringBuilder sb, int i, int j) {
    char temp = sb.charAt(i);
    sb.setCharAt(i, sb.charAt(j));
    sb.setCharAt(j, temp);
}
```

reverse() 参数是 start 和 end，表示要反转的字符范围。start 是起始索引，end 是结束索引（不包含 end）。

```
private void reverse(StringBuilder sb, int start, int end) {
    while (start < end) {
        swap(sb, start, end);
        start++;
        end--;
    }
}
```

- swap() → 交换两个字符
- reverse() → 调用 swap() 来反转字符串
- solution() → 调用 reverse() 来反转字符串

? for (int[] query : queries)

- for-each 遍历 queries 数组，每个元素是 int[] 类型的数组。

? sb.setCharAt(i, sb.charAt(j));

- 在 Java 中，StringBuilder 的 setCharAt() 方法用于设置指定索引处的字符。

```
sb.setCharAt(i, sb.charAt(j));
```

方法	说明
sb.charAt(j)	返回索引 j 处的字符 (char)
sb.setCharAt(i, ...)	将索引 i 处的字符设置为 ... (char)

在代码中，我们使用 sb.charAt(j) 来获取索引 j 处的字符，并使用 sb.setCharAt(i, ...) 来设置索引 i 处的字符。

```
StringBuilder sb = new StringBuilder("hello");  
sb.setCharAt(1, sb.charAt(4)); // 'e' → 'o'  
System.out.println(sb); // 输出: "hollo"
```

setCharAt(i, sb.charAt(j)) 交换 i 和 j 位置的字符。这通常用于实现字符串反转，因为反转需要交换首尾字符、次尾字符，依此类推。通过 (swap) 交换 temp 和当前元素。

Revision #4

Created 13 July 2025 14:31:16 by Dain

Updated 13 July 2025 14:43:47 by Dain