











Largest Number / My Java Solution to share

My Java Solution to share







Same code by user Daring with comments: https://leetcode.com/discuss/21488/java-code-by-providingcomparator-with-explaination-nlogn







Compare between the string needs O(len) and sort needs O(len*log(len)), the time complexity is O(len^2*log(len))







may be Integer.toString(num[i]) is better than num[i] + ""







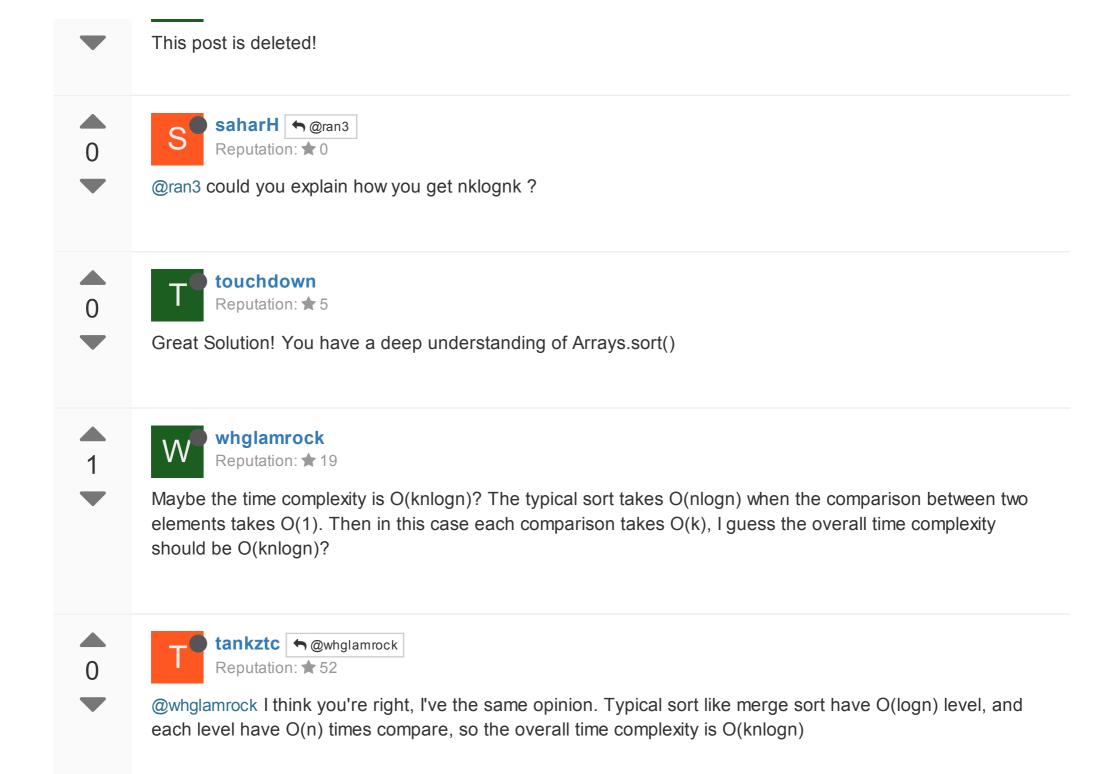
I would recommend using the following comparator if you wish to reduce memory footprint, since the comparator above generates a lot of concatenated strings (of the order of O(n²), with n the total number of elements). Also for large-size input arrays, the following comparator will have a smaller chance to trigger GC, which is detrimental to the time performance.

```
Comparator<String> cmp = new Comparator<String>() {
        @Override
        public int compare(String str1, String str2) {
                sb1.delete(0, sb1.length()).append(str1).append(str2);
                sb2.delete(0, sb2.length()).append(str2).append(str1);
                for (int i = 0; i < sb1.length(); i++) {
                        if (sb1.charAt(i) == sb2.charAt(i)) continue;
                        return (sb1.charAt(i) > sb2.charAt(i) ? -1 : 1);
                return 0;
};
```

I assume you have initialized two final StringBuilder objects sb1 and sb2 somewhere before the comparator.







19 10587

POSTS **VIEWS** Log in to reply