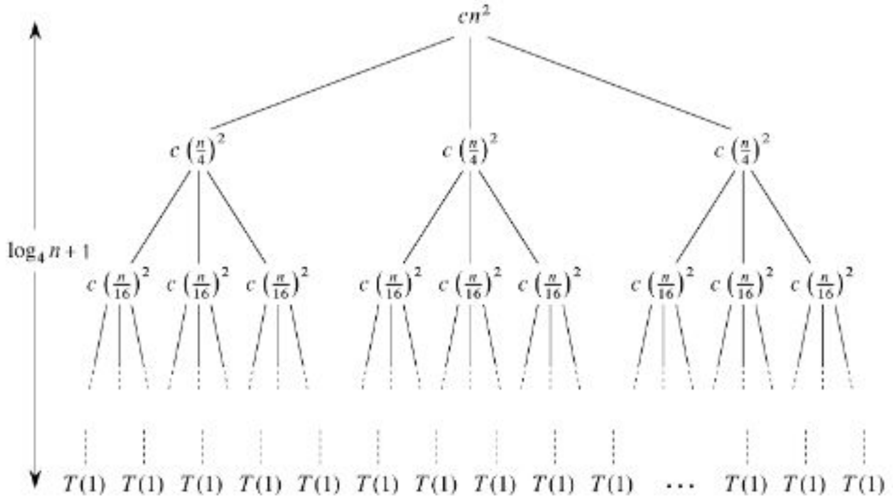
Group 5

(Sorting competition 2021)

By: Noah and Cole

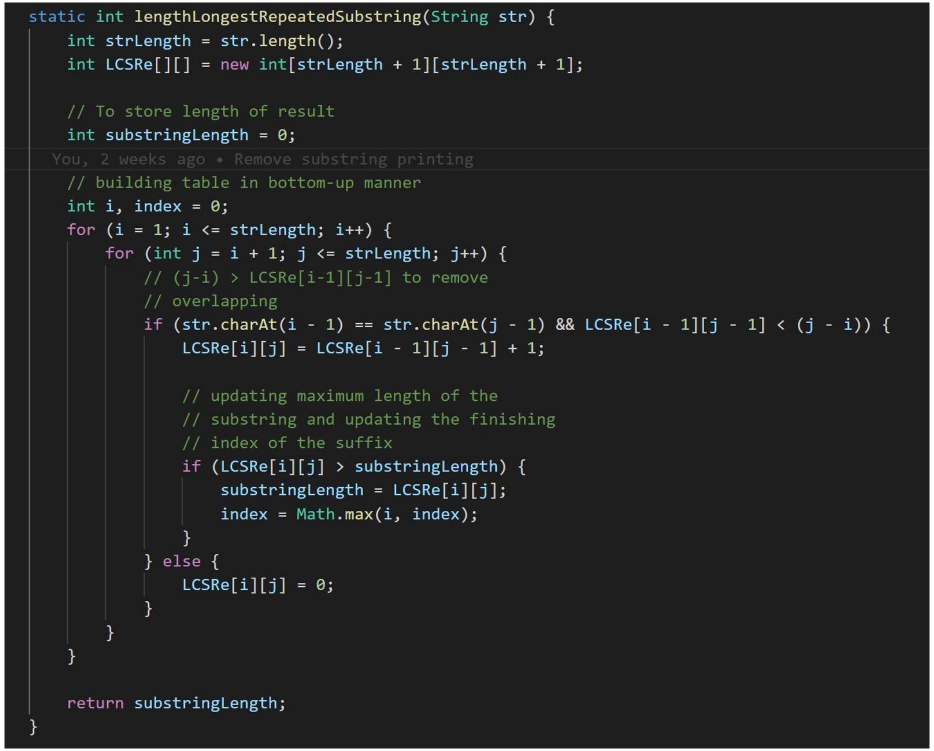
Times

- Round 1:
  - Run Time: 2571ms
- Round 2:
  - Run Time 4554ms
- No Correctness issues were reported!
- 8th place overall



Modifications (Helper5)

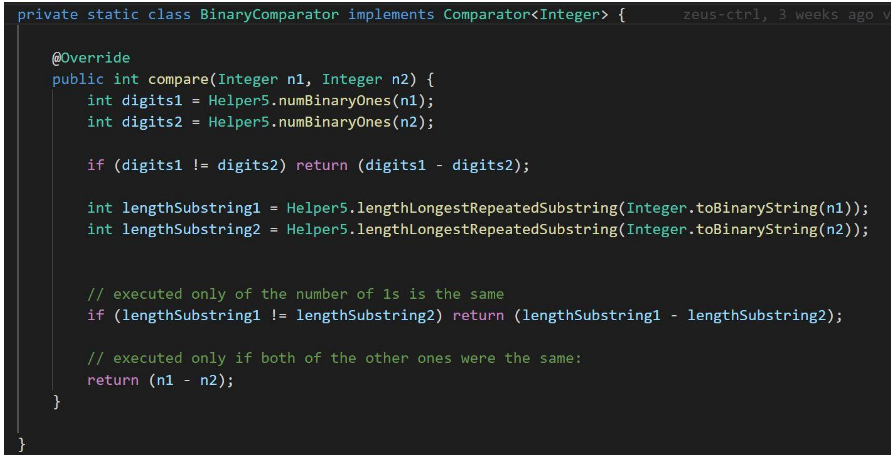
- Data stored as an Array of Integer Objects
- Improved the efficiency of finding the longest repeating and non-overlapping substring
    - Old Time Complexity: O(n^4)
    - New Time Complexity: O(n^2)



```java
static int lengthLongestRepeatedSubstring(String str) {
    int strLength = str.length();
    int LCSRe[][] = new int[strLength + 1][strLength + 1];

    // To store length of result
    int substringLength = 0;
  You, 2 weeks ago • Remove substring printing
    // building table in bottom-up manner
    int i, index = 0;
    for (i = 1; i <= strLength; i++) {
        for (int j = i + 1; j <= strLength; j++) {
            // (j-i) > LCSRe[i-1][j-1] to remove
            // overlapping
            if (str.charAt(i - 1) == str.charAt(j - 1) && LCSRe[i - 1][j - 1] < (j - i)) {
                LCSRe[i][j] = LCSRe[i - 1][j - 1] + 1;

                // updating maximum length of the
                // substring and updating the finishing
                // index of the suffix
                if (LCSRe[i][j] > substringLength) {
                    substringLength = LCSRe[i][j];
                    index = Math.max(i, index);
                }
            } else {
                LCSRe[i][j] = 0;
            }
        }
    }

    return substringLength;
}
```

Modifications (Group5)

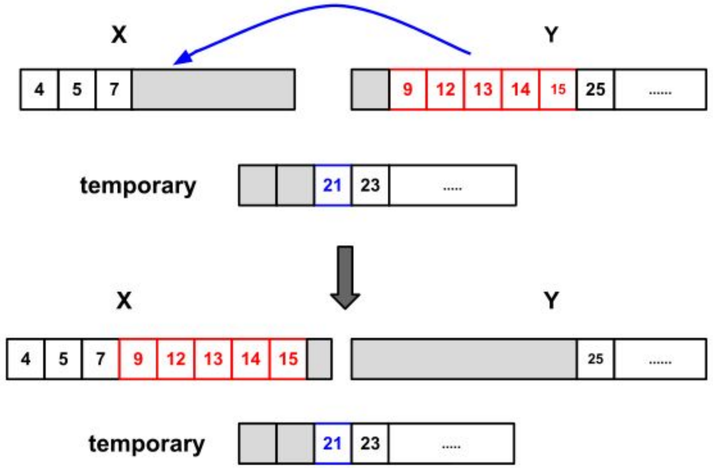
- Move up the conditional that compares the digits
- Prevents extraneous calls to lengthLongestRepeatedSubstring method
- Array is then Sorted using java.util.Arrays sort method which sorts Integer Objects using TimSort



```java
private static class BinaryComparator implements Comparator<Integer> {      zeus-ctrl, 3 weeks ago

    @Override
    public int compare(Integer n1, Integer n2) {
        int digits1 = Helper5.numBinaryOnes(n1);
        int digits2 = Helper5.numBinaryOnes(n2);

        if (digits1 != digits2) return (digits1 - digits2);

        int lengthSubstring1 = Helper5.lengthLongestRepeatedSubstring(Integer.toBinaryString(n1));
        int lengthSubstring2 = Helper5.lengthLongestRepeatedSubstring(Integer.toBinaryString(n2));


        // executed only of the number of 1s is the same
        if (lengthSubstring1 != lengthSubstring2) return (lengthSubstring1 - lengthSubstring2);

        // executed only if both of the other ones were the same:
        return (n1 - n2);
    }

}
```

Run Time

- TimSort (Java default for sorting Objects)
  - Expected: O(n log(n))
  - Worst: O(n log(n))
- Our Comparator
  - Expected: O(m^2/2)
  - Worst: O(m^2)
- Our Comparator with TimSort
  - Expected: O(m^2/2 * (n log(n)))
  - Worst: O(m^2 * n log(n^2))



Questions?