1   BinaryST

- If an element is not in the tree, then its frequency is 0, rank is -1

- Consider the multi-set $\{A, A, B, B, B\}$. Rank of an element x is defined as number of elements less than x. There are 2 elements (A, A) that are less than B. Thus rank of B is 2. There are no elements that are less than A, thus rank of A is 0.

- If an element appears multiple times, then it should appear multiple times in both `inOrder` and `preOrder` traversals.

- Time taken for constructor (non-default), `inOrder`, `preOrder` can not be $O(h)$. It should be proportional to number of elements in the set.

- Time taken for other methods: It should $O(h)$ comparisons, not $O(h)$ time. Note that time taken for each comparison depends on the size of the strings that are being compared.

- Say the multi-set is $\{A, A, A, A\}$. What should be the height? It depends on your implementation.

- You should design your data structure so that all the queries can be answered as efficiently as possible. The only constraint in your design is that you are not allowed to balance the tree.

2   War Story

- You are given S which the set of all k length substrings of (unknown) U. Note that S is a set, not a multi-set.

- Your goal is to compute all **possible** $2k$-length substrings of U. This needs to be computed from S. Note that it is not possible to compute the exact set containing all $2k$-length substrings of U (without knowing U). Thus T will be a set consisting of all possible $2k$ length strings that could be inferred from S.

- Please see the text for an example.

- Their are typos in constructor names: They should match with class names.

- For 2.2, specs say that S should be stored only as a BST; However, in addition you could store S as an `array/arary list`.

- For 2.3, specs say that S should be stored only as hash sets/hash table; However, in addition you could store S as an `array/array list`.

- You can design your own implementation of hash set/hash table or use Java's inbuilt classes. If you use Java's inbuilt classes, please read the documentation of these classes so that you are familiar with intricacies of the functionality of these classes.

- Do not try to construct U from S.