2. size -> int length();// TC: O(1)
3. convert to char Array -> toCharArray(); // TC: O(n)
4. value for specific index -> charAt(int index); // TC: O(1)
5. substring from string -> substring(a,b] // a : inclusive, b: Exclusive, TC: O(n)
6. transform to Lowercase -> toLowerCase(); // TC: O(n)
7. transform to UpperCase -> toUpperCase(); // TC: O(n)
8. replace all characters in string -> replaceAll(from, to) // TC: O(n)
9. Some useful Character properties

- Character.isLetter();
- Character.isAlphabetic();
- Character.isUpperCase();
- Character.isLowerCase();
- Character.isDigit();

10. Concatenation

- T str1 + str2
- StringBuilder ->
    - new StringBuilder() / new StringBuilder(int)
    - append("adding string") // better way to do
    - toString() // converting back to string

7. HashSet // Collection

1. Definition ->Set set = new HashSet<>();
2. insert / update -> boolean add(t); // TC: O(1)
3. delete -> boolean remove(t); // TC: O(1)
4. get -> boolean contains(t); // TC: O(1)
5. size -> int size(); // TC: O(1)
6. check for Empty -> boolean isEmpty(); // TC: O(1)
7. remove all set values -> clear(); // TC: O(n)

Helpful? *Let me know if i missed anything*
Update: Please read comment section too, people are awesome :)
Please upvote!
Happy Coding :)

Helpful link for Time Complexity(TC).

- https://stackoverflow.com/questions/7294634/what-are-the-time-complexities-of-various-data-structures
- https://www.baeldung.com/java-collections-complexity

string array java map time-complexity syntax coding-style data structure

Comments: 18 Best Most Votes Newest

Type comment here... (Markdown is supported)