Brady, Harland, Sam

The Results of RALtester3.java

“testing routine for RaggedArrayList

insertion order: qwertyuiopasdfghjklzxcvbnmaeiou

The number of comparison to build the RaggedArrayList = 234

TEST: after adds - data structure dump

[0] -> [a][a][b][c][d][ ][ ][ ]

[1] -> [e][e][f][g][h][i][ ][ ]

[2] -> [i][j][k][l][m][ ][ ][ ]

[3] -> [n][o][o][p][q][ ][ ][ ]

[4] -> [r][s][t][ ]

[5] -> [u][u][v][w][x][y][z][ ]

[6] -> null

[7] -> null

STATS:

list size N = 31

square root of N = 5.568

level 1 array 6 of 8 used.

level 2 array sizes: min = 3 used, avg = 5.2 used, max = 7 used

Successful search: min cmps = 2, avg cmps = 31.7, max cmps 62

TEST: contains("c") ->true

TEST: contains("7") ->true

TEST: toArray

[a][a][b][c][d][e][e][f][g][h][i][i][j][k][l][m][n][o][o][p][q][r][s][t][u][u][v][w][x][y][z]

TEST: iterator

[a][a][b][c][d][e][e][f][g][h][i][i][j][k][l][m][n][o][o][p][q][r][s][t][u][u][v][w][x][y][z]

TEST: sublist(e,o)

[0] -> [e][e][ ][ ]

[1] -> [f][g][ ][ ]

[2] -> [h][i][ ][ ]

[3] -> [i][j][ ][ ]

[4] -> [k][l][m][n][ ][ ][ ][ ]

[5] -> null

[6] -> null

[7] -> null”

Worst time cases.

1. contains():
    - Worst-case time complexity: O(N)
    - Reason: In the worst case, we might need to scan every element in every L2Array across the entire RaggedArrayList, which results in linear time complexity.
2. Iterating through the whole list:
    - Worst-case time complexity: O(N)
    - Reason: Iterating through the entire list would require visiting every element in each L2Array. Since the list holds N elements in total, iterating through them will take O(N)O(N)O(N).
3. toArray():
    - Worst-case time complexity: O(N)
    - Reason: Copying every element from the RaggedArrayList to the array requires visiting each element exactly once, so the time complexity is linear in terms of the number of elements N.
4. subList():

- Worst-case time complexity: O(M, where M is the number of elements in the sublist.
- Reason: The method traverses from fromElement to toElement and copies all elements in that range into a new RaggedArrayList. The number of operations depends on how many elements are between the two given elements, which is MMM.

Bugs

There are no know bugs in our code.