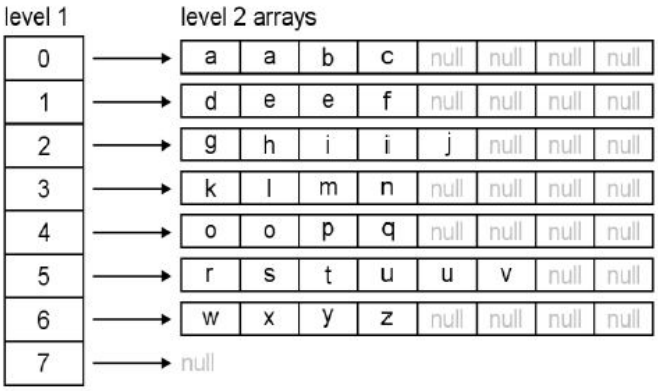
CSCI-290 *Project Part 5*

Ragged Array List - Complete

Objectives: Objectives: Building a new data structure (the rest of the methods).

Now we are going to write the rest of the methods for the Ragged Array List: contains(), toArray(), subList(), and the iterator.



Setting up the testing code:

1. On Brightspace you will find a new testing file:
   - RALtester3.java
2. Copy this into the student package of your project.
3. This is just a java file so you won't need a run configuration. Just select it and run it.
4. It will add 31 single character strings to your ragged array list as shown in the picture, but your list may be different because of different choices when adding. The tester tests each of the new methods and also calculates some performance statistics.

Task 1: Implement contains(item)

- Use findFront() it already does the search.
- You can run the testing code as you implement each one of these tasks. Unlike the previous testing code this just prints the results. You'll have to make sure they are correct.

Task 2: Implement the Iterator class.

- The iterator() method just creates an instance of the private inner class Itr. Itr implements the Iterator interface. You only need to implement the methods hasNext() and next(). Use the provided implementation of remove() to satisfy the compiler.
- Internally the iterator keeps its position as a ListLoc. Implement the ListLoc method moveToNext() which moves the ListLoc to the next item in the list.
- hasNext() needs just 1 line of code
- next() took me 6 lines. Remember to throw an exception if the user goes past the end of the list.

Task 3: Implement toArray(a)

- This is the version of toArray() where the caller passes in an array of the correct size. You should check for the correct size, but you don't have to reallocate a larger array or null out the excess locations like the collection classes do.
- I used an iterator to simplify writing this.

Task 4: Implement subList(E fromElement, E toElement)

- The subList() method returns a new independent RaggedArrayList whose elements range from fromElement, inclusive, to toElement, exclusive. For example subList(b, e) on the above example should return [b, c, d]. The original list is unaffected, but both lists refer to the same data objects.
- I used findFront(), ListLoc.equals(), ListLoc.moveToNext(), the RaggedArrayList() constructor and add().

Task 5: Analysis On a raggedArrayList with N items, what are the worst case times for:

1. contains()
2. iterating through the whole list
3. toArray()
4. subList()

Explain and justify your answers.

What to turn in:

Written part:

1. Include all member names and indicate which one did the electronic submit.
2. Print the result of running RALtester3.java to a pdf in your project folder.
3. Your analyses from Task 5.
4. Any incomplete parts, or any known bugs in your code.

Electronic Submission

- I will create groups and put you into them. Anyone in the group can upload files.
- Print your write up to pdf name it Project5.pdf be sure that the names of the team members are in the write up on the top of first page.
- RaggedArrayList.java every single method that was not written by Bob Boothe, should have an author indicated in the method header.
- Upload these two files to Brightspace. MAKE sure that the team member names are in the writeup!! Do not zip anything.

Grading:

Written part – report and test results	10 points
Task 1 contains()	10 points
Task 2 iterator()	20 points
Task 3 toArray()	20 points
Task 4 subList()	20 points
Task 5 analyses of execution times	20 points

I expect your programs to be properly commented and use good style. Points may be deducted for egregious disregard of these matters. Every method in every class must have complete documentation including authorship. A full an complete revision history should be at the top of every class file!