Chilembwe Asante

Midterm Exam

Spacco

220

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Tries**. Write a *recursive* method that counts the number of leaf nodes (i.e. nodes with no children) in a Trie. This doesn’t require a helper method.

| public int countLeafNotes() { |
| --- |

static int countLeafNodes()

{

int result = 0;

if (this.children.isEmpty())

result++;

for (char c : children.keySet()) {

if (check.children.isEmpty())

result += wordCount(root.children[i]);

return result;

}

1. **Tries, part 2**. Write a method takes an int parameter depth and returns the number of nodes (word or non-word nodes) that are deeper than the given depth. Feel free to use a recursive helper method.

public int getNumDeeperThanHelper(Trie nodesearch, int result) {

for (char c : nodesearch.children.keySet()) {

count++;

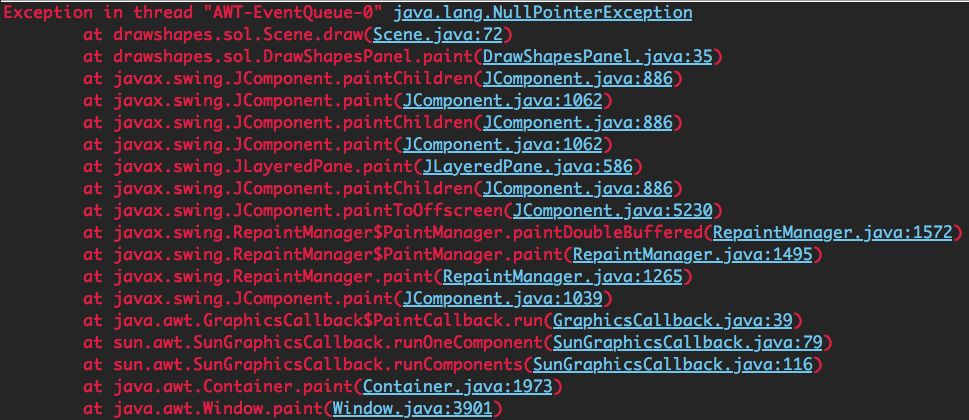
}

return result;

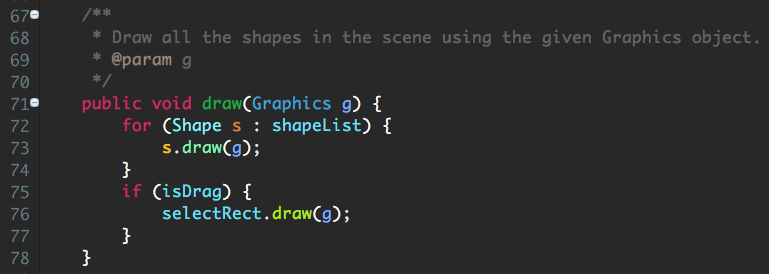
}

| public int getNumDeeperThan(int depth) {  int count = 0;  Trie nodesearch = this;  if (depth == 0 && !children.isEmpty()) {  for (char c : children.keySet()) {  nodesearch = children.get(c);  count +== getNumDeeperThanHelper(nodesearch, result);  }  } else if (depth == 0 && children.isEmpty()) {  count = 0;  return result;  } else if (depth != 0) {  for (char c : children.keySet() {  nodesearch = children.get(c);  count += checknode.getNumDeeperThan(depth-1); |
| --- |
| }  }  return result;  } |
|  |

1. **Debugging.** Suppose you get the following stack trace:



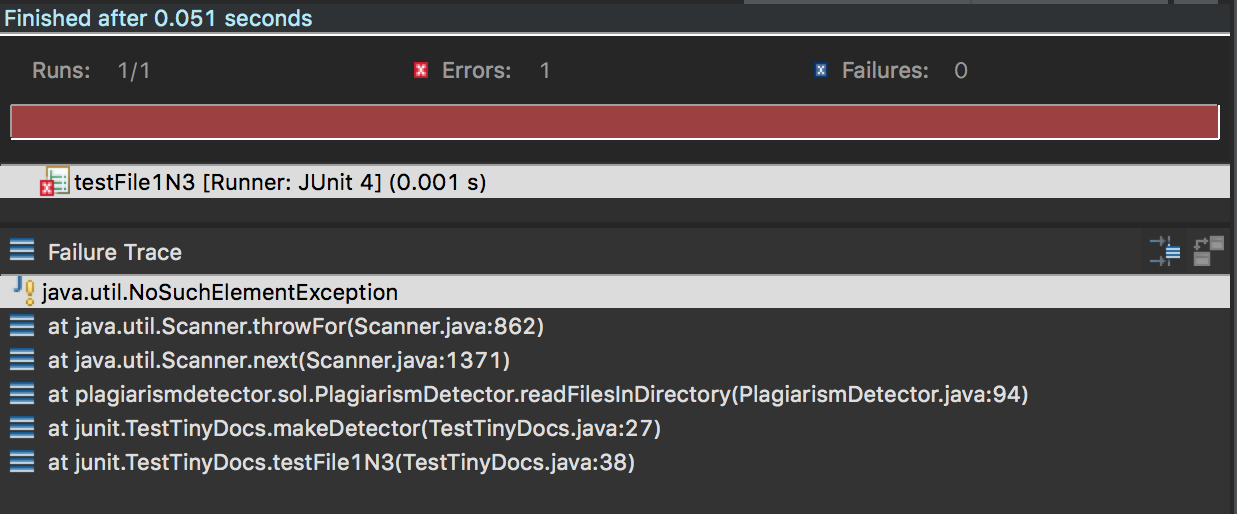
And it seems to indicate a problem in this method:



What is wrong? Be specific! There is enough information provided here to indicate exactly what is wrong.

I believe the error is regarding to the draw method for the code. The error is flagging a null. This means that the draw method does not have the correct shapes implemented and it is unable to call them so it is coming up as null.

1. **More debugging.** Suppose you run one of your test cases from the Plagiarism Detector assignment, and you get the following stack trace:



* 1. Which line should you double click first, and why?

PlagairismDectector.java:94

This would have all of the code written in it. I think starting here would allow you to overlook the majority of the errors.

* 1. Which line or lines should you double click second, and why?

The two test cases (TestTinyDocs) at the bottom of the list. This is the place where you implement the test cases so you could see how they work.

* 1. Which line or lines should you not double click, and why?

The util.Scanner lines because they are built into the Java code as is. Usually this is very fragile and the slightest error can mess up the entire code.

1. **Legendary debugging stories.** What is one of the most challenging bugs you’ve fixed this term? Describe what was wrong, approximately how long you worked on it, and what eventually fixed it.

The worst issue was getting the bounding box to work for all the shapes in the lab. The hardest part was getting the dimensions of the shape perfect to surround the shape. I realized that x,y was (0,0) at the top left. Once I realized this I was able to use parameters for given shapes to surround the shapes appropriately.

1. **Plagiarism Detection** Four computer science students, Alicia, Beatrice, Carlos, and DeAndre, are trying to figure out if there was plagiarism in a set of 6 documents. They write a program that computes the number of n-grams in common between all pairs of documents.

The 6 documents, d1 to d6, have the following numbers of n-grams in them (n=4):

| **Document** | **n-grams** |
| --- | --- |
| **d1** | 954 |
| **d2** | 12,134 |
| **d3** | 911 |
| **d4** | 935 |
| **d5** | 11,127 |
| **d6** | 982 |

Now suppose that the following grid represents the number of n-grams in common between each pair of documents.

|  | **d1** | **d2** | **d3** | **d4** | **d5** | **d6** |
| --- | --- | --- | --- | --- | --- | --- |
| **d1** | --- |  |  |  |  |  |
| **d2** | 51 | --- |  |  |  |  |
| **d3** | 57 | 49 | --- |  |  |  |
| **d4** | 48 | 57 | 43 | --- |  |  |
| **d5** | 52 | 599 | 50 | 62 | --- |  |
| **d6** | 39 | 49 | 52 | 51 | 48 | --- |

Alicia, Beatrice, Carlos, and DeAndre then examine their results, but they reach very different conclusions.

* Alicia thinks that these results tell us almost nothing about possible plagiarism.
* On the other hand, Beatrice thinks that d2 and d5 probably contain plagiarism.
* Meanwhile, Carlos thinks that these results mean that there was probably no plagiarism.
* Finally, DeAndre thinks that every pair of documents probably contains plagiarism with at least one other document.

Who do you think is right? Explain why you agree or disagree with each perspective.

I agree with Alicia in that the results given don’t really show any evidence of plagiarism.

I do not agree with Beatrice. D2 and d5 have 599 n-grams in common. If we say that d2 and d5 total n-gram count is 12,000, they also share 5% of the content. This will surely happen with a n-gram count of 4 and having comparison between the two. An increase in the count can be modified for change in size of d2 and d5.

I agree with Carlos. There might not have been plagiarism but we aren’t really sure. SInce there is no access to the documents they used.

1. **Spam classification and common words.** Suppose that we have a training set of 100,000 messages, 75,000 of which have been classifier as Spam and 25,000 of which have been classifier as not Spam, or Ham.
   1. Alicia implements a spam classifier, and computes that P(Spam | the) is 0.5 (i.e. the probability that a message is Spam given that the message contains the word “the” is 0.5). Beatrice thinks this value cannot possibly be right. Who do agree with, and why?

I agree with Alicia. I believe that the value will almost be accurate.

* 1. Carlos wants to use the total number of times a word occurs in their calculations, rather than simply the number of messages that contain the word at least once. So they rewrite the code so that P(word) is the number of times a word occurs across all messages, including duplicates, is:

DeAndre points out that this could produce nonsensical values. Why does DeAndre think this? Give a specific example.

For example, a word that appeans n times in b ham messages but 0 spam. This would mess up a value when you’re trying to predict how probable it is that a word will appear in a specific category.

1. Suppose that our team (Alicia, Beatrice, Carlos, DeAndre) are studying tests for a very rare disease. Overall, P(Sick) is 1/2000 (one out of two thousand people), and the current best test has a false positive rate of 50% (i.e. P(+ | H)) and a false negative rate of 10% (i.e. P(- | S)).
   1. Alicia and Carlos read about a clinical trial for a new test for this disease. The results of the clinical trial for the new test are listed below. Compare the false positive and false negative rates of this new test with the previous false positive and false negative rates of the previous test. Is it better or worse?

P(+ | H) = .16 ( - | S) = .00012

The results are better.

|  | Sick | Healthy | Total |
| --- | --- | --- | --- |
| Positive test (+) | 10 | 2 | 12 |
| Negative test (-) | 1 | 8117 | 8118 |
| Total | 11 | 8119 | 8130 |

* 1. Beatrice and DeAndre are extremely skeptical of the reliability of the results of this clinical trial. Why might they be skeptical?

This is not the most reliable result. There aren’t many results to have a diverse motion.