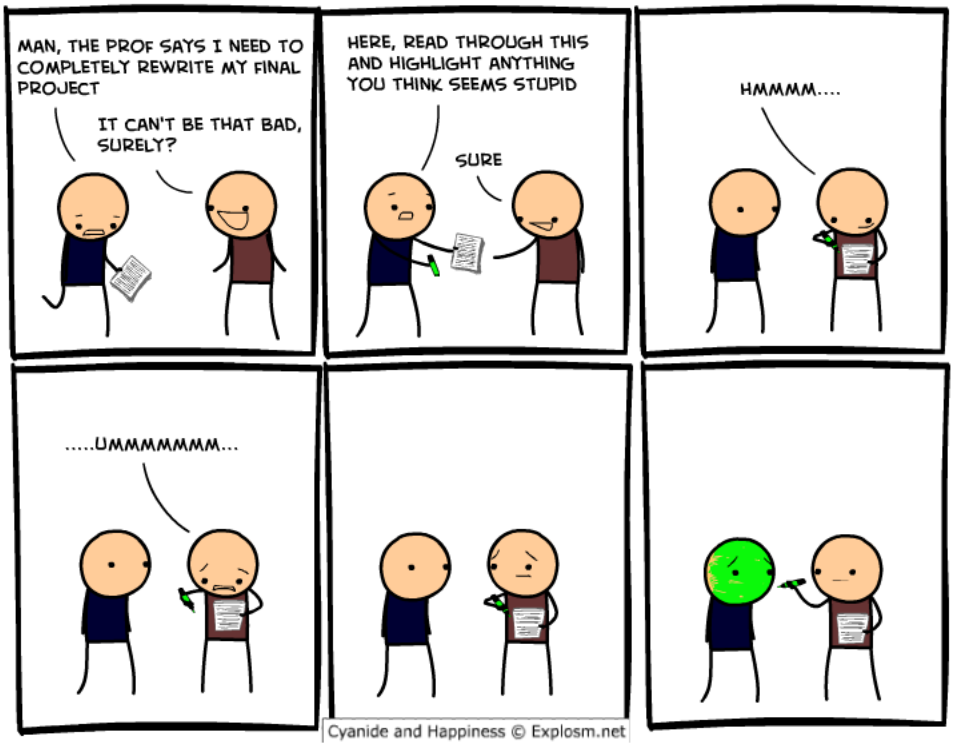
CS201 - Assignment 10



http://explosm.net/comics/2083/

For our final assignment you are to implement a data structure that we haven't implemented yet in this class and run some experiment that examines the performance of that data structure.

Read through this entire handout so that you understand what is required of you and when the deadlines are.

You may (and are encourage to) work with a partner on this assignment.

Acceptable Resources

Almost all data structures you might be interested in investigating have very, very likely been implemented by someone else (in fact, probably by, many, many people). You may look at any resources in the book or online that give algorithmic descriptions of the problem. However, **you may not look at or use any code from the book or online**.

If you're having trouble finding good resources, come by and talk to me.

Part 1: Proposal

Due: Friday, May 2 at the beginning of class

By this Friday, you need to decide what data structure you're going to implement. Even though what you're submitting for this part is minimal (see below) make sure to spend a few hours looking into the different options. In particular, make sure you understand a bit of the complexity of what is involved in implementing the data structure. You will be graded based on the difficulty of what you decided to do.

Here are some ideas for things you might consider implementing:

- a linked list suite including: singly linked list with just a head reference, singly linked list with a head and tail reference, circularly linked list and doubly linked list with head and tail reference. These would each be separate classes, but should all implement the same interface.
- a balanced tree. Some options include:
  - red-black tree
  - 2-3-4 tree
  - splay tree
  - B-tree
- Skew heap
- Binomial heap
- Faster than $O(n \log n)$ sorting: radix sort and/or bucket sort
- Hashtable concepts, e.g.
  - Explore a few different hash functions
  - Explore a few different open address schemes
- Disjoint sets
- Something else you've heard of!

Once you've decided, submit a .txt file with the following information:

- You name(s)
- What you plan to implement. Be specific!
- A citation for one resource you plan to use. It could be a website, book, paper, etc.

Part 2: Implementation and Testing

Due: Wednesday, May 7 at the beginning of class

- Your data structure must support generics. If it needs a restriction in the generic (like `comparable`), that's fine.
- Try and be consistent with naming methods. If we've implemented another variation of this data structure (e.g. binary trees) consider using the same names.
- Your implementation must be appropriately efficient. For example, if you implement a balanced binary search tree, the run-times of the methods should be consistent with what you'd expect for a balanced binary search tree.
- You must write JUnit tests to test all of your public methods.

Part 3: Writeup

Due: Friday, May 9 at 6pm

Once you've implemented the data structure, I want you to run an experiment examining the performance of the data structure. Write a 1 page writeup that includes:

- Your name
- What data structure you implemented
- Experimental results for one or two of the "important" features of the data structure.
- You must include at least one table or graph
- A short analysis of your experimental result that describe what you should theoretically expect to see from the result (e.g. big-O run-time of the method) and how you can see that in the experiments.

You may submit your writeup as either a .txt or .pdf file.

Grading

You will be graded on:

- How well you followed the specifications above
- The correctness of your implementation

- The difficulty/ambitiousness of your project
- The quality of your tests
- Code style and documentation
- The quality of your writeup