

# Algorithm Analysis and Design (CS1.301)

## Project Proposal Functional Implementation of Algorithms

Abhinav S Menon

### Project Plan

The project will consist of an implementation of all the algorithms covered in class in Haskell (a functional programming language).

### Motivation

The pseudocode for all the algorithms shown in class was based on imperative languages. However, functional code tends to significantly differ from the imperative implementation in most cases. The major features of imperative programs which are absent in functional languages are:

- global variables
- loops
- side effects
- variable re-assignment

All these features are extensively made use of in the pseudo-code shown in class, which means that a functional implementation of the same algorithms will require a considerably different perspective to code.

Therefore, coding the algorithms in a functional language like Haskell will, I feel, improve my understanding of them by providing an alternate perspective. It will also habituate me to writing functional code, which is a useful skill as functional programs are frequently much shorter<sup>1</sup> than equivalent imperative programs.

---

<sup>1</sup>Hudak, Paul, and Mark P. Jones. Haskell vs. Ada vs. C++ vs. awk vs.... an experiment in software prototyping productivity. Technical report, Yale University, Dept. of CS, New Haven, CT, 1994.

## Use Case

As mentioned above, since functional code is usually smaller than other implementations, it is possible to make use of these programs to a compact software that requires an efficient implementation of them.