**Language Preference:** F#

**Group Members**: Thomas Nix, Michael Peddycord, Chuong Vu, Toubee Lo, Stephen Meadley

Optimum Sorting Application

We plan on building an application in one of the present languages that has multiple forms of sorting operations and can demonstrate efficiency in sorting algorithms. These types of sorting are fundamental in any language and a critical part of Computer Science that put emphasis on design patterns, O(n) efficiency, and data types. Examples of these types of sorts would be bubble sort, merge sort, and selection sort. The idea of including algorithms such as a convex hull algorithm have come up, but we are unsure if this is something we would like to integrate as well.

Since these languages all incorporate parallel processing, this would be an interesting topic as far as efficiency goes. The idea is that we could play around with some of the functionality of parallel processing to further express efficiency in sorting, i.e comparing single thread performance to multi threaded performance, or something of the like. This is part of the reason we aren’t sure about including convex hull as part of the application, because that level of complexity may be overkill and too demanding based on the other things we want to include.

There are a few papers or resources that we should read that we have so far found online, including the online F# guide, the MSDN entry for parallel and async design patterns in F#, and an entry on fsharpforfun.com comparing F# to C#, so that we can understand fundamental differences compared to a language that is somewhat more familiar (all of these are listed below)

Web Programming with F#:

<http://fsharp.org/guides/web/>

Async and Parallel Design Patterns in F#: Parallelizing CPU and I/O Computations:

<https://blogs.msdn.microsoft.com/dsyme/2010/01/09/async-and-parallel-design-patterns-in-f-parallelizing-cpu-and-io-computations/>

Comparing F# with C#: Sorting:

<https://fsharpforfunandprofit.com/posts/fvsc-quicksort/>