Parallel Functional Programming Project Proposal

Hsuan-Ru Yang (hy2713)

I plan to implement a solver for the "Minimum Spanning Tree Problem".

Description of the problem (from Wikipedia):

A minimum spanning tree of a connected graph is a subset of the edges that forms a tree that includes every vertex, where the sum of the weights of all the edges in the tree is minimized.

One algorithm that solves this problem is "Kruskal's algorithm". The algorithm can be described as:

1. Create a forest F (a set of trees), where each vertex in the graph is a separate tree
2. Create a set S containing all the edges in the graph
3. While S is nonempty and F is not yet spanning
    - Remove an edge with minimum weight from S
    - If the removed edge connects two different trees then add it to the forest F, combining two trees into a single tree

At the termination of the algorithm, the forest has a single component and forms a minimum spanning tree.

I will try to understand Haskell's parallel functions and improve the algorithm.

References:

https://en.wikipedia.org/wiki/Minimum_spanning_tree
https://en.wikipedia.org/wiki/Kruskal%27s_algorithm