

Information

I struggled quite a bit with this project. I spent many days trying to parse the .dat files, understand what data I wanted to keep, and what data I could throw away. I was mystified about how I could visit every city, even though some cities had no trip back to SLO. In the spec, it says that a return trip is infinitely large, so visiting a city without a return route would result in an infinitely long distance.

After speaking with other students (yesterday), I found that many teams were creating return trips for every city, resulting in an undirected graph. I think that I could have made much more progress if I could have done that from the beginning. I spent a lot of time creating an algorithm to determine whether or not any city had a return trip back to SLO, only to find that I didn't need to use it.

I also didn't know that data structures are very difficult to offload to the Xeon Phi. I would have attempted to use only arrays had I known this earlier.

I resulted to creating a return path for every city for my TSP algorithm. I used a greedy approach (Nearest Neighbor) to traverse the cities and calculate the distance. I combined all airports in a single city. This means that I gathered all of the cities that all airports in a city can fly to, and put them in a data structure that I named "City." This ensures that I don't prune cities that have an exit from another airport in the same city. I wasn't able to offload my solution because I have many complicated data structures.

Some of the methodologies that I was planning to use were to create different threads to deploy python scripts to parse the input files. I was also trying to discover an effective data structure that provided with data locality. This would have consisted of some kind of adjacency matrix that mapped to neighboring cities of each source city.

Tools

I used a templated library called "minicsv" to parse the .dat files into my data structures. It was difficult to find a library that supported the operations that I required and are able to compile on the lab machines. I would have used openMP to offload my code if I were able to get it to work.

In hindsight

- I would have spent less time trying to optimize the parsing of the csv files, because I didn't know that we could have modified the files and provided you with them to run with our solutions.
- I would have used much more simple data structures so that I could offload them. I wasn't able to offload to the Xeon Phi because I don't know how to offload my data structures.
- I would have pruned cities with no return trips or created return trips for cities that don't have one. I spent too much time trying to find a solution to the TSP problem without modifying the input.